



Escola d'Enginyeria de Telecomunicació i  
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TREBALL FINAL DE GRAU

**TÍTOL DEL TFG: Mecanismes de coordinació en xarxes SDN amb múltiples controladors**

**TITULACIÓ: Grau en Enginyeria Telemàtica**

**AUTOR: Oriol Gresa Solsona**

**DIRECTOR: Anna Agustí Torra**

**DATA: 24 d'octubre del 2018**

**Títol:** Mecanismes de coordinació en xarxes SDN amb múltiples controladors

**Autor:** Oriol Gresa Solsona

**Director:** Anna Agustí Torra

**Data:** 24 d'octubre del 2018

## Resum

El ràpid creixement del món d'Internet i el gran ventall d'avantatges que aporta a la vida de la gent ha provocat que l'actual arquitectura de xarxa s'hagi vist sobrecarregada arribant a certs nivells que excedeixen de les possibilitats que ofereix.

Aquest fet ha generat un replantejament de la xarxa i intentar convertir-la en una xarxa més escalable amb major rapidesa i menor cost. Això s'està intentant dur a terme amb la transformació de les xarxes cap a un entorn virtual i canviant el maquinari per hardware no propietari, desacoblant el pla de dades del pla de control on un controlador (o conjunt de controladors) passa a gestionar tota la Intel·ligència de la xarxa.

Tecnologies com SDN són les que ens permeten fer aquest desacoblament, i malgrat que encara estan en ple desenvolupament, ja es comencen a veure certes millores respecte a les xarxes anteriors. Tant és així que algunes companyies telefòniques ja utilitzen aquestes tecnologies.

Entre els projectes que s'estan portant a terme en aquest àmbit, es troba el projecte ONOS que ja s'ha convertit en un dels controladors SDN open source més utilitzats per la creació de solucions de les tecnologies mencionades.

L'objectiu d'aquest projecte es centra en la implementació d'una aplicació que permeti controlar la gestió dels dispositius entre els controladors de la xarxa en les versions 1.14 i 1.15 d'ONOS.

Com a resultat, s'ha aconseguit crear dos aplicacions per coordinar els dispositius entre els controladors, tant per la versió 1.14 com per la 1.15, i a part s'ha creat una documentació de tots els passos seguits per arribar a crear aquestes aplicacions, per a un futur poder seguir implementant aplicacions per controlar la gestió dels dispositius segons els criteris desitjats amb unes guies a seguir per tal de ser més eficients i no perdre massa temps en la comprensió de l'estructura i funcionament del codi.

**Title:** Coordination mechanisms in SDN networks with multiple controllers

**Author:** Oriol Gresa Solsona

**Advisors:** Anna Agustí Torra

**Date:** October 24th 2018

## Overview

The fast growth of Internet and all the advantages that contribute to people's life has caused that the current network architecture has been overloaded reaching some levels that exceed of the possibilities that it offers.

This fact has generated a reconsideration of the network and try to turn it into a more scalable and reliable network, by a faster way and less cost. This is trying to carry out with the transformation of the networks to some virtual environment and changing the hardware for non proprietary hardware, decoupling the data layer from the control layer where a controller (or set of controllers) manages all the Intelligence of the network.

Technologies like SDN are those that allow us to do this decoupling, and in spite of that still are in full development already, it begins to see some improvements with regard to the previous networks. So that, there's some telephonic companies that already use these technologies.

Among the projects that are being carried out in this field, we find the ONOS project that has already become one of the open source SDN controllers most used for the creation of solutions for the aforementioned technologies.

The aim of this project focuses on the implementation of an application that allows to manage the devices among the network controllers in ONOS versions 1.14 and 1.15.

As a result, it has been possible to create two applications to coordinate the devices between the controllers, both for version 1.14 and for 1.15, and in addition a documentation has been created of all the steps followed to get to create these applications, for in the future, to be able to continue implementing applications to manage the devices according to the criteria desired with some guidelines to be followed in order to be more efficient and not to lose too much time in understanding the structure and functioning of the code.

# ÍNDEX

<b>INTRODUCCIÓ .....</b>	<b>1</b>
<b>CAPÍTOL 1. INTRODUCCIÓ A ONOS .....</b>	<b>3</b>
1.1 SDN .....	3
1.2 L'Open Netowrk Operating System (ONOS) .....	4
1.3 Conceptes bàsics d'ONOS .....	4
1.3.1 CLI d'ONOS .....	5
1.3.2 GUI d'ONOS .....	8
1.3.3 Clúster .....	10
1.3.4 Metodologia de creació de noves aplicacions .....	12
<b>CAPÍTOL 2. ESCENARIS BASATS EN ONOS .....</b>	<b>15</b>
2.1 Fitxers a modificar .....	15
2.2 Versió 1.14 .....	16
2.2.1 topologia utilitzada .....	16
2.2.2 Verificació del correcte funcionament de les noves aplicacions .....	17
2.3 Versió 1.15 .....	22
2.3.1 topologia utilitzada .....	22
2.3.2 Configuracions addicionals per la versió 1.15 .....	24
2.3.3 Verificació del correcte funcionament de les noves aplicacions .....	27
<b>CAPÍTOL 3. CONCLUSIONS .....</b>	<b>32</b>
<b>BIBLIOGRAFIA.....</b>	<b>34</b>
<b>ACRÒNIMS .....</b>	<b>35</b>
<b>ANNEXOS.....</b>	<b>36</b>
ANNEX A. INSTAL·LACIÓ D'ONOS .....	36
ANNEX B. INSTAL·LACIÓ D'ATOMIX .....	38
ANNEX C. INSTAL·LACIÓ DE BAZEL .....	40
ANNEX D. CONTINGUT DELS FITXERS A MODIFICAR DE L'APLICACIÓ MOVE-DEVICE	41
ANNEX E. CONTINGUT DELS FITXERS A MODIFICAR DE L'APLICACIÓ BALANCING....	44

# INTRODUCCIÓ

Internet és, avui en dia, una eina imprescindible per bona part de la població del primer món i segons dades disponibles a [1] s'estima que l'any 2018 ja és més del 50% de la població mundial la que té accés a Internet. L'era de la digitalització està transformant la manera de relacionar-se, d'interactuar i de treballar i està desencadenant un canvi radical en moltes empreses i en els seus clients. El concepte de digitalització va més enllà de la utilització del correu electrònic i de la navegació web i engloba també la necessitat d'estar permanentment connectat amb tot i amb tothom, facilitant la possibilitat de poder obtenir informació o de poder-se comunicar amb qualsevol en temps real.

Aquesta necessitat de connectivitat permanent i d'immediatesa, juntament amb la rapidesa amb la que avança la innovació tecnològica, ha posat de manifest la incapacitat que té l'arquitectura de xarxa existent per satisfer els requeriments de les noves tecnologies i del nous serveis. Per tant, cal un paradigma de xarxa que permeti un funcionament àgil i flexible dels recursos, i que tingui la capacitat d'adaptar-se al desplegament de nous serveis i aplicacions de banda ampla (IPTV, VoD, videotrucades) i de reajustar-se per permetre la incorporació de nous dispositius intel·ligents (IoT).

Aquest canvi de mercat és un gran repte i una clara oportunitat de negoci pels proveïdors de serveis, que es veuen obligats a repensar l'arquitectura de la xarxa i a buscar solucions que permetin adaptar-se a les noves necessitats de forma ràpida i rentable, avançant cap a xarxes més flexibles, escalables i eficients. El paradigma Software Defined Networks (SDN)[2] persegueix definir solucions per a aquest nou plantejament de l'arquitectura de la xarxa.

SDN separa el pla de dades del pla de control, de tal manera que hi ha un element, el controlador SDN, que s'encarrega de configurar i controlar els diferents elements de xarxa. Per tal d'evitar centralitzar la funció de control en un únic element, es preveu la formació de clústers de controladors que gestionen de forma conjunta i coordinada els dispositius de la xarxa.

En aquest treball s'ha estudiat el concepte de virtualització de xarxa, s'ha experimentat amb el controlador de SDN anomenat Open Network Operating System (ONOS)[3] i s'ha treballat en la definició i desplegament d'escenaris SDN controlats per un clúster de controladors, amb l'objectiu principal de crear una aplicació que permeti modificar o definir quin controlador del clúster és el que s'encarrega de cada element de la xarxa.

El document està dividit en els capítols que es descriuen a continuació:

- Capítol 1: Introducció a ONOS. En aquest capítol s'expliquen alguns conceptes bàsics per familiaritzar-se amb ONOS.
- Capítol 2: Escenaris basats en ONOS. En aquest capítol es testeja les aplicacions que proporcionen les versions 1.14 i 1.15, i s'implementen dos aplicacions addicionals.

- **Conclusions i línies futures.** En aquest capítol es comenten les principals conclusions a les que s'ha arribat en la realització d'aquest projecte i es plantegen possibles línies futures.

Adicionalment s'han inclòs 5 annexos d'instal·lacions de software i modificacions de fitxers necessàries per la realització del treball.

- Annex A: Instal·lació d'ONOS.
- Annex B: Instal·lació d'Atomix[4].
- Annex C: Instal·lació de Bazel[5].
- Annex D: Contingut dels fitxers a modificar de l'aplicació move-device.
- Annex E: Contingut dels fitxers a modificar de l'aplicació balancing.

## CAPÍTOL 1. INTRODUCCIÓ A ONOS

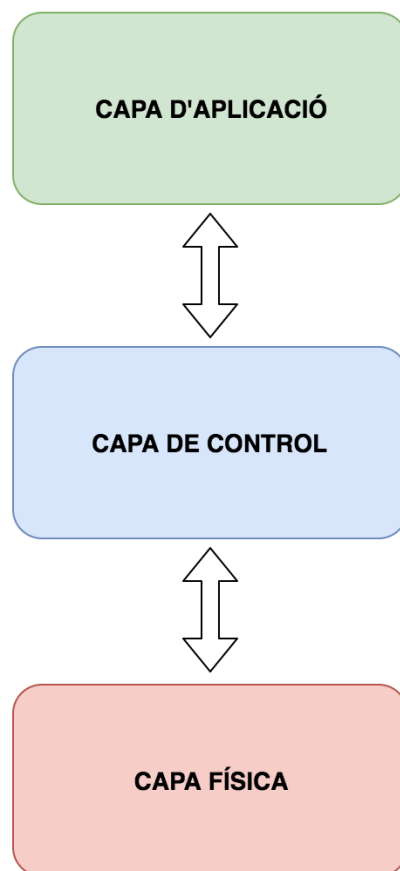
En aquest capítol s'expliquen els conceptes bàsics de les xarxes SDN, es presenta el controlador de xarxa ONOS i es descriuen les eines d'aquest controlador que s'han fet servir en aquest treball.

### 1.1 SDN

Les xarxes definides per software (en anglès Software Defined Networks, SDN) són un nou paradigma de l'arquitectura de la xarxa que facilita la gestió, permet una configuració més eficient i programable per millorar el rendiment i la supervisió de la xarxa. Estan pensades per fer front a l'arquitectura estàtica de les xarxes tradicionals, que són complexes i descentralitzades, i convertir-la en una arquitectura més flexible i centralitzant la intel·ligència de la xarxa en un sol element, el controlador.

Es basen en desacoblar el pla de dades del pla de control, que es converteix en programable. Aquesta característica permet simplificar considerablement la infraestructura de la xarxa, permetent l'ús de dispositius de menor cost.

Les xarxes SDN estan formades per tres capes diferents:



**Fig 1.1** capes de l'arquitectura SDN.

- Capa física: està formada per la infraestructura de la xarxa (hosts, switches i mitjans de transmissió).
- Capa de control: està formada pel controlador (o conjunt de controladors), que té una visió global, i incorpora el sistema operatiu de la xarxa. És on es prenen les decisions i és on es programen les taules de flux dels elements de la capa física a través de les SouthBound APIs. La funció bàsica d'aquestes APIs és permetre al controlador comunicar-se amb els elements de commutació de la xarxa i programar la lògica de les comunicacions al hardware.
- Capa d'aplicació: està formada per les aplicacions dels usuaris i és des d'on es controla el comportament de la xarxa a través de les NorthBound APIs. Aquestes APIs incorporen els patrons d'utilització de la xarxa de cada aplicació i la seva funció és comunicar aquests patrons a la capa de control, on es prendran les decisions que caldrà aplicar a la capa física.

## 1.2 L'Open Network Operating System (ONOS)

El sistema operatiu Open Network Operating System (ONOS) és un projecte nascut l'any 2014 i desenvolupat per la Open Network Foundation (ONF)[6] i per un gran seguit d'entitats i empreses que col·laboren en el projecte. La finalitat d'ONOS és desenvolupar un controlador SDN basat en la creació d'un sistema de codi obert (open source) que permeti virtualitzar la part troncal de la xarxa.

El projecte és dinàmic i s'actualitza periòdicament, publicant noves versions del codi cada 4-5 mesos. Actualment s'està desenvolupant la versió 1.15.

## 1.3 Conceptes bàsics d'ONOS

Les dues eines d'ONOS que s'han fet servir en aquest projecte són: la interfície de línies de comandes (Command Line Interface, CLI) i la interfície gràfica d'usuari (Graphical User Interface, GUI). En aquest apartat es descriuen les seves característiques i funcionalitats.

A més a més, com que l'objectiu és desenvolupar una eina que pugui modificar l'assignació dels dispositius físics dins d'un clúster de controladors, també es descriu el concepte de clúster SDN i s'explica la metodologia de creació de noves aplicacions.



### 1.3.1 CLI d'ONOS

El CLI (Command Line Interface) d'ONOS[7] és una eina que permet la interacció per línia de comandes amb ONOS, ja sigui per visualitzar aspectes de la configuració, de dispositius connectats, activació/desactivació d'aplicacions, accedir als logs d'ONOS, etc.

Per accedir-hi s'ha d'executar la comanda **onos @IP** on @IP és la direcció IP on està executant-se ONOS. Alternativament, també es pot utilitzar la comanda **onos localhost** sempre i quan s'executi a la màquina que es vol gestionar.

En cas de tenir algun dubte de funcionament de les aplicacions, si s'escriu la comanda i s'afegeix l'opció **--help** s'obté tota la informació de l'aplicació així com els paràmetres que requereix.

```
[onos> apps --help
DESCRIPTION
    onos:apps

    Lists application information

SYNTAX
    onos:apps [options]

OPTIONS
    -a, --active          Show active only
    -j, --json            Output JSON
    -n, --name            Sort by application ID name
    -s, --short           Show short output only
    --help               Display this help message

onos> █
```

**Fig 1.2** exemple d'utilització opció --help.

Com s'observa a la figura anterior, primerament apareix una descripció de l'aplicació, a continuació la sintaxi correcta per utilitzar-la i per últim les opcions que accepta l'aplicació.

Les comandes d'ús més freqüent es descriuen a continuació:

- Nodes:

La comanda "nodes" s'utilitza per saber tots els nodes als quals està connectat el node que s'està gestionant. Dit d'una altra forma, s'observa tots els

controladors ONOS connectats al controlador que es gestiona (aquest també es mostra, i està identificat amb un asterisc “\*” al final de la línia).

```
[onos> nodes  
id=192.168.60.23, address=192.168.60.23:9876, state=READY, version=1.14.0.06d0a77, updated=48m44s ago *  
onos> █
```

**Fig 1.3** Resultat comanda “nodes” amb un sol controlador.

```
[onos> nodes  
id=192.168.60.23, address=192.168.60.23:9876, state=READY, version=1.14.0.06d0a77, updated=4m30s ago *  
id=192.168.60.24, address=192.168.60.24:9876, state=READY, version=1.14.0.06d0a77, updated=4m22s ago  
id=192.168.60.25, address=192.168.60.25:9876, state=READY, version=1.14.0.06d0a77, updated=4m22s ago  
onos> █
```

**Fig 1.4** Resultat comanda “nodes” en un clúster de tres controladors.

Com s’observa a les figures anteriors, a la primera només apareix un sol controlador, i a la segona n’apareixen tres. Això es degut a que abans d’executar la comanda s’ha creat un clúster (concepte explicat al punt 1.3.3) de tres controladors. Les dades que s’obtenen són l’identificador del node, que és la seva direcció IP, la direcció IP especificant també el port on atén peticions, en quin estat es troba, la versió d’ONOS que utilitza i per últim el temps que porta d’execució.

- Masters:

La comanda “masters” s’utilitza per saber els switches que gestiona cada node de la xarxa, identificats per la seva direcció MAC.

```
[onos> masters  
192.168.60.23: 6 devices  
  of:0000000000000001  
  of:0000000000000002  
  of:0000000000000003  
  of:0000000000000005  
  of:0000000000000006  
  of:0000000000000007  
192.168.60.24: 0 devices  
192.168.60.25: 1 devices  
  of:0000000000000004  
onos> █
```

**Fig 1.5** Resultat comanda “masters” en un clúster de tres controladors.

Com s’observa a la figura anterior, apareixen els tres controladors del clúster i els switches que gestiona cadascun d’ells.

- Summary:

La comanda “summary” s'utilitza per obtenir un resum del node. S'especifica la direcció IP del node, la versió d'ONOS instal·lada, el nom del clúster al qual forma part (en el cas que el node no formi part d'algun clúster amb d'altres nodes, s'interpreta que forma part d'un clúster on només hi ha aquest node), el nombre de nodes que formen el clúster, i el nombre de switches, links, hosts, flows i intents<sup>1</sup> de la xarxa.

```
onos> summary
node=192.168.60.23, version=1.14.0.06d0a77 clusterId=default
nodes=3, devices=7, links=12, hosts=0, SCC(s)=1, flows=28, intents=0
onos>
```

**Fig 1.6** Resultat comanda “summary” en un clúster de tres controladors.

Com s'observa a la figura anterior, apareix la direcció IP del node, la versió d'ONOS instal·lada, el nom del clúster al qual pertany, el nombre de nodes del clúster, i el nombre de dispositius, links, flows i intents de la xarxa.

- Apps -a -s:

La comanda “apps -a -s” s'utilitza per saber les aplicacions actives (gràcies a l'opció -a) i amb una descripció curta (gràcies a l'opció -s, sense aquesta opció s'obté molta més informació de les aplicacions) que té ONOS.

```
onos> apps -a -s
* 23 org.onosproject.drivers          1.14.0.SNAPSHOT Default Drivers
* 46 org.onosproject.optical-model    1.14.0.SNAPSHOT Optical Network Model
* 52 org.onosproject.openflow-base    1.14.0.SNAPSHOT OpenFlow Base Provider
* 61 org.onosproject.hostprovider      1.14.0.SNAPSHOT Host Location Provider
* 64 org.onosproject.lldpprovider      1.14.0.SNAPSHOT LLDP Link Provider
* 65 org.onosproject.openflow          1.14.0.SNAPSHOT OpenFlow Provider Suite
* 89 org.onosproject.mobility          1.14.0.SNAPSHOT Host Mobility
* 96 org.onosproject.fwd               1.14.0.SNAPSHOT Reactive Forwarding
* 97 org.onosproject.proxyarp          1.14.0.SNAPSHOT Proxy ARP/NDP
onos>
```

**Fig 1.7** Resultat comanda “apps -a -s”.

Com s'observa a la figura anterior, apareixen les 9 aplicacions actives. L'asterisc indica que l'aplicació està activa (si no s'utilitzés l'opció “-a” a l'hora d'executar la comanda, apareixerien totes les aplicacions instal·lades i les que tenen l'asterisc són les actives), el número que identifica cada aplicació, el paquet al qual pertany, la versió de cada aplicació i per últim el nom.

<sup>1</sup> Ruta fixa entre dos dispositius de xarxa.

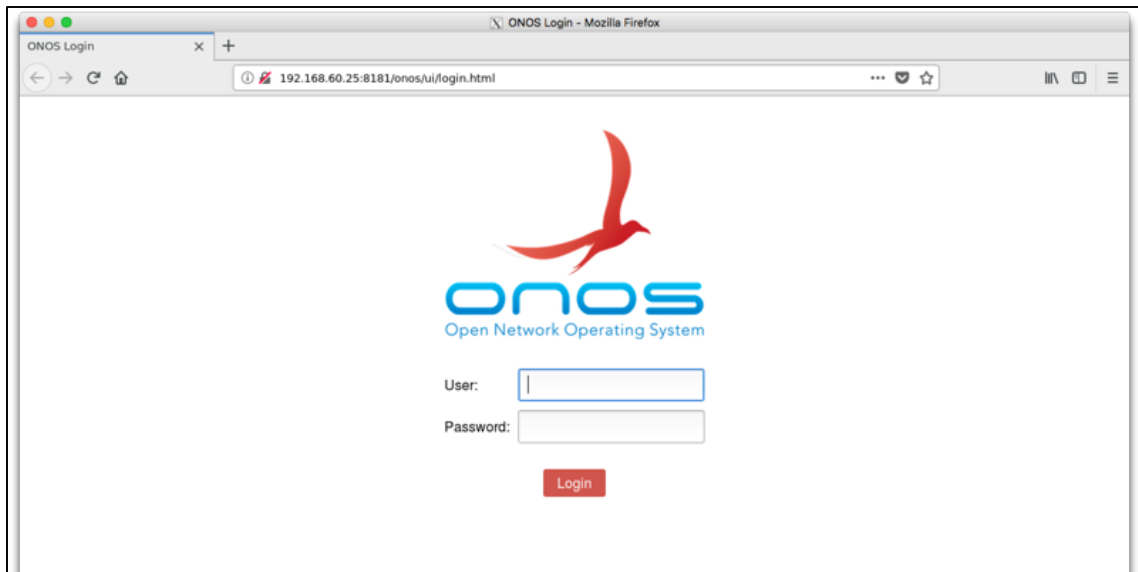
- Log:display:

La comanda “log:display” s'utilitza per obtenir tots els logs que ha generat ONOS fins al moment en que s'executa. Alternativament hi ha la comanda “log:tail” que mostra el mateix que “log:display”, però enlloc de tornar-nos al CLI, segueix mostrant els logs futurs d'ONOS fins que es decideix parar la comanda amb “Ctrl + C”. En aquest moment es torna al CLI.

### 1.3.2 GUI d'ONOS

El GUI (Graphical User Interface) d'ONOS[8] és una eina que ens permet la interacció gràfica amb ONOS, on s'observen els controladors que formen part del clúster identificats amb diferents colors, la seva IP i el nombre de dispositius al seu càrrec, la topologia de la xarxa amb els switches del color del controlador que els gestiona, i un resum amb la mateixa informació que s'obté a l'executar la comanda “summary” al CLI.

Per accedir-hi s'ha d'obrir un navegador i introduir la següent adreça: **http://XX.YY.WW.ZZ:8181/onos/ui**, on XX.YY.WW.ZZ és l'adreça IP de la màquina on s'executa ONOS. Al accedir a aquesta adreça es mostra el següent:

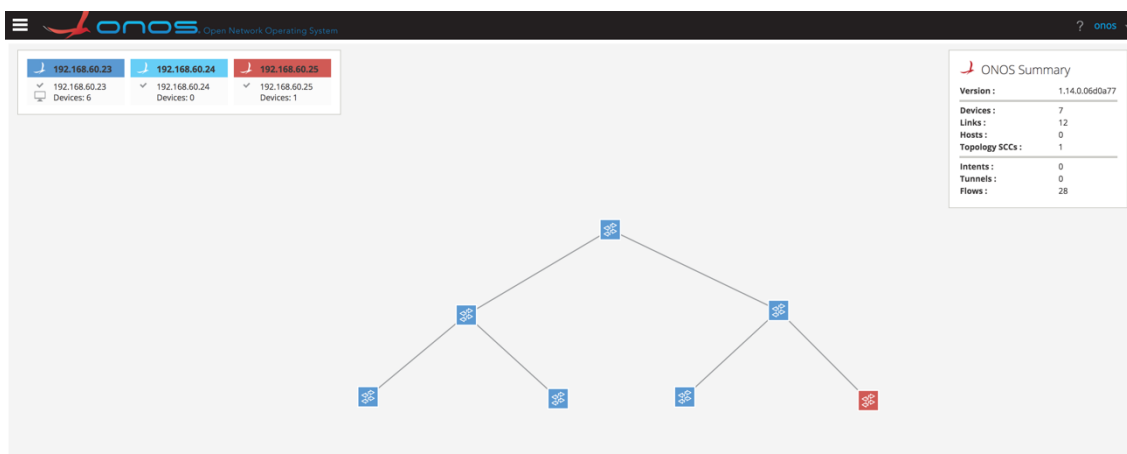


**Fig 1.8** Pàgina de login d'ONOS.

Per tal d'accedir, les credencials són:

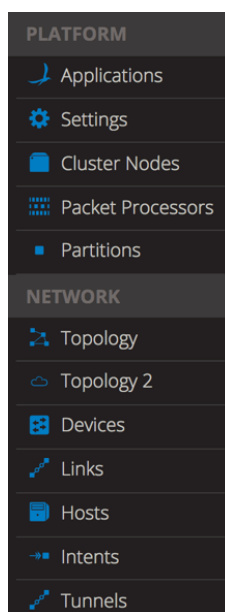
- User: onos
- Password: rocks

Una vegada s'han introduït les credencials d'accés, apareix la següent pantalla:



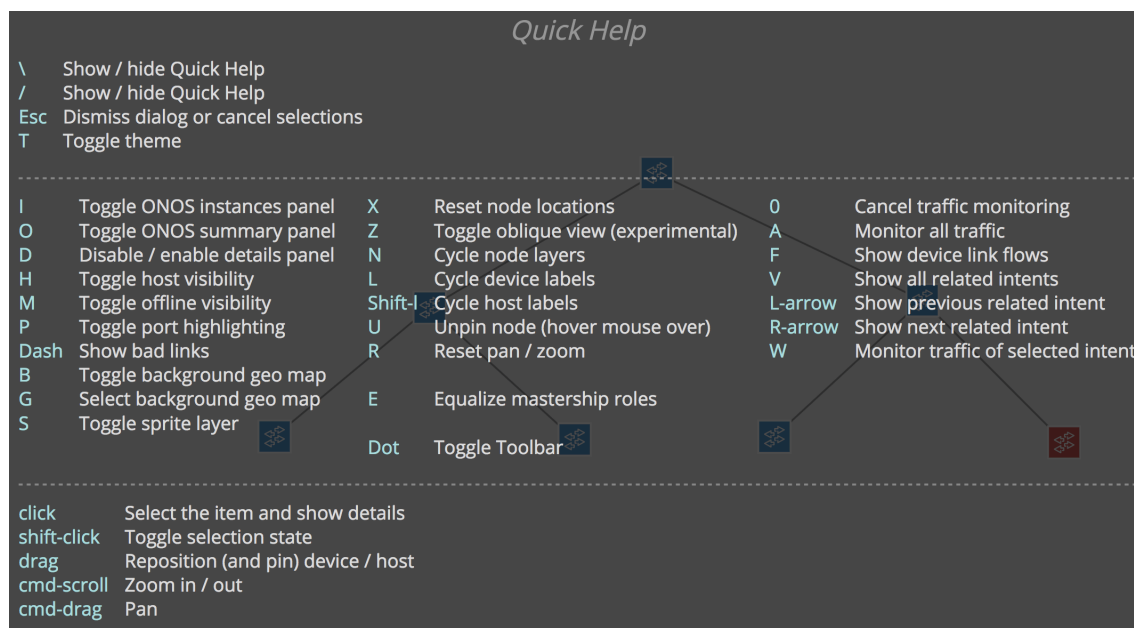
**Fig 1.9** Pàgina del GUI d'ONOS.

Com s'observa a la figura anterior, a la part superior esquerra apareixen els controladors que formen el clúster, cadascun identificat d'un color diferent, a la part dreta apareix el summary, i a la part central apareix la topologia de la xarxa amb els switches del mateix color que el controlador que el gestiona. A més a més, a dalt de tot a l'esquerra hi ha tres ratlles horitzontals blanques per desplegar un menú amb diverses opcions.



**Fig 1.10** Menú desplegable del GUI d'ONOS.

Adicionalment, el GUI proporciona un menú d'ajuda amb les tecles configurades per realitzar un seguit d'accions. Per accedir-hi cal estar dins del GUI i prémer la lletra "ç" del teclat (pot variar segons el teclat de l'ordinador. Proves fetes amb un MacBook Pro amb teclat espanyol).



**Fig 1.11** Menú d'ajuda del GUI d'ONOS.

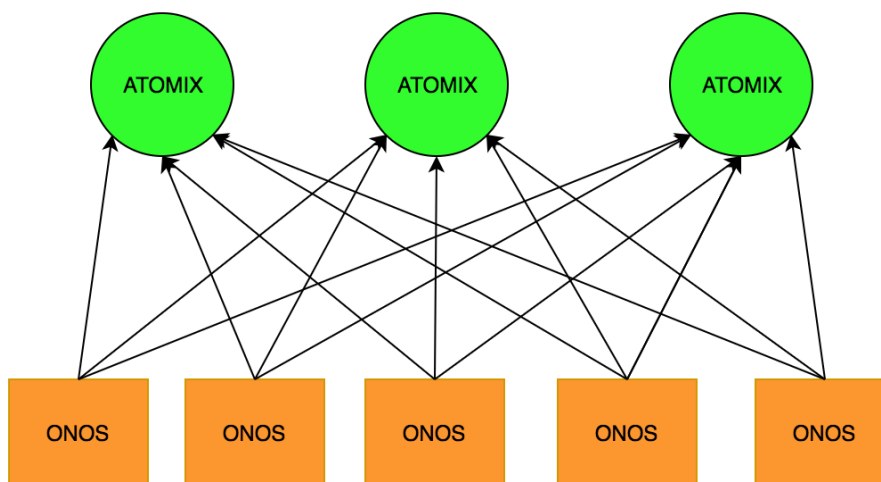
### 1.3.3 Clúster

En ONOS, un clúster s'entén com un conjunt d'instàncies/nodes d'ONOS interconnectades lògicament entre elles i s'utilitza per evitar tenir un únic punt de fallada del sistema. En les xarxes SDN el controlador és el punt des d'on es gestiona tot el pla de control de la xarxa, i si aquest falla, queda afectada tota la xarxa. És per aquest motiu que al combinar varies instàncies es disminueix la probabilitat que la xarxa pateixi fallades.

En el moment de començar amb aquest projecte, l'última versió estable que hi havia d'ONOS era la versió 1.13 i ja s'estava començant a desenvolupar la versió 1.14. En aquestes versions la gestió del clúster es fa de forma local[9], executant un script des de línia de comandes, i al moment en que es genera el clúster ja no s'hi pot unir cap node més.

Per a què un node s'uneixi s'ha de destruir el clúster existent i crear-ne un de nou. El mateix passa quan un node s'apaga, si es vol tornar a unir al clúster se n'ha de crear un de nou. Aquest funcionament és poc pràctic ja que no és un model escalable, i per aquest motiu a les últimes actualitzacions de la versió 1.14 (que actualment ja hi ha una versió estable) i a la nova versió 1.15 s'ha canviat el funcionament del clúster[10].

En aquestes noves actualitzacions s'ha externalitzat el servei del clúster a uns nous nodes (Atomix).

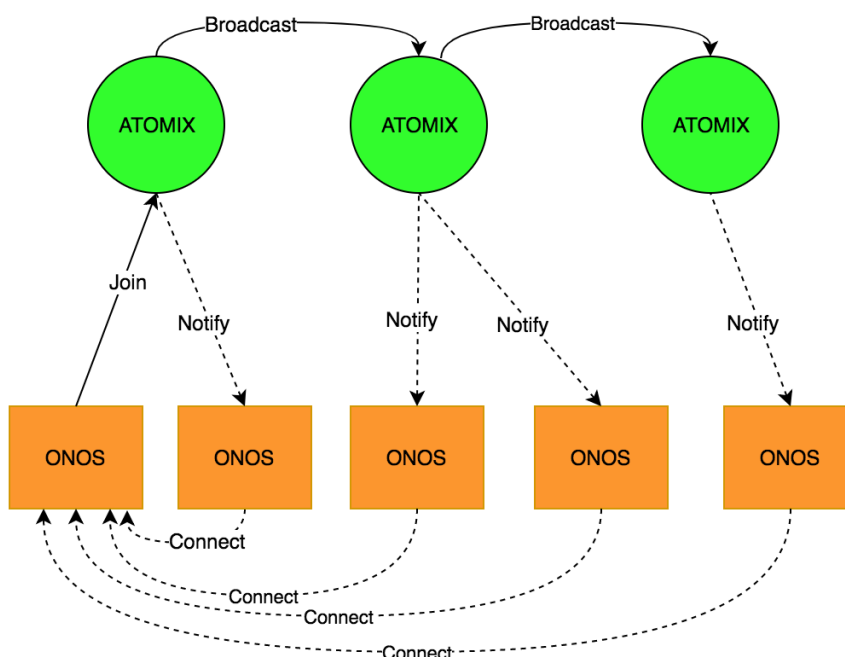


**Fig 1.12** Exemple nova arquitectura del clúster.

Aquests nodes són els encarregats d'atendre les peticions dels nodes d'ONOS d'unir-se al clúster i d'informar a la resta de nodes quan algun d'ells es connecta/desconnecta.

Gràcies a aquest nou procés es poden unir i desconnectar els nodes sense la necessitat de destruir i posteriorment crear un nou clúster cada vegada. L'únic que necessiten els nodes d'ONOS és conèixer la ubicació dels nodes Atomix.

Normalment es recomana tenir el mateix nombre de nodes Atomix que de nodes ONOS (si es pretén tenir 3 nodes ONOS s'haurien d'utilitzar 3 nodes Atomix, però no hi ha cap problema si hi ha més nodes d'ONOS que d'Atomix com es mostra de forma teòrica a les figures 1.12 i 1.13).



**Fig 1.13** Procés que es segueix quan s'uneix un nou node.

### 1.3.4 Metodologia de creació de noves aplicacions

Dins d'ONOS es poden trobar dos tipus d'aplicacions. Les que s'executen des del CLI i les que s'executen automàticament de forma periòdica. A continuació s'explica quina és la metodologia de creació de noves aplicacions per a cada cas.

#### 1.3.4.1 Aplicacions del CLI

Per a que una aplicació nova es pugui executar des del CLI en qualsevol de les versions actuals d'ONOS s'ha de declarar la nova aplicació dins del directori **onos/cli**, i implementar les funcions que l'aplicació ha de cridar dins del directori **onos/core**.

Per la part de la declaració de la nova aplicació dins del directori **onos/cli** s'han de realitzar dos modificacions. Per un costat dins de **onos/cli/src/main/java** es crea un fitxer de nom **AppNameCommand** on s'hi defineix el nom de l'aplicació (nom a partir del qual s'executa), una breu descripció i per últim la crida a les funcions implementades dins del **core**. Per un altre costat dins de **onos/cli/src/main/resources/OSGI-INF/blueprint/shell-config.xml** es declara la nova aplicació creada.

```

1  //...//
16 package org.onosproject.cli;
17
18 import ...
19
20
21 //...//
22 Forces device mastership rebalancing.
23 //...//
24 @Command(scope = "onos", name = "balance-masters",
25           description = "Forces device mastership rebalancing")
26 public class BalanceMastersCommand extends AbstractShellCommand {
27
28     @Override
29     protected void execute() {
30         get(MastershipAdminService.class).balanceRoles();
31     }
32
33 }

```

Fig 1.14 Exemple de fitxer de definició d'aplicació.

```

<command>
  <action class="org.onosproject.cli.BalanceMastersCommand"/>
</command>

```

Fig 1.15 Exemple de declaració d'aplicació.



### 1.3.4.2 Aplicacions periòdiques

Per a que una nova aplicació es pugui executar periòdicament s'ha de declarar dins del directori **onos/apps**, i definir dins del fitxer de mòduls el nom del seu OAR (ONOS Application aRchive)

Per la declaració de la nova aplicació dins del directori **onos/apps** s'han de realitzar quatre modificacions. Primer es crea l'arbre de directoris de la nova aplicació (**onos/apps/AppName/src/main/java/org/onosproject/AppName/**), i a continuació dins d'aquest nou directori safegeixen dos fitxer amb nom **package-info.java**, on s'hi defineix el package de la nova aplicació, i **AppNameCommand.java**, on s'hi defineix el temps de periodicitat i la crida a les funcions implementades dins de **onos/core**.

Per últim, en funció de la versió d'ONOS que s'utilitzi s'han de realitzar les dos modificacions següents:

Per la versió 1.14 o posteriors dins del directori **onos/apps/AppName/** es crea un fitxer **BUCK** que conté els paràmetres de compilació, i dins del fitxer **onos/modules.defs** es defineix el OAR de la nova aplicació.

Per la versió 1.15 dins del directori **onos/apps/AppName** es crea un fitxer **BUILD** que conté els paràmetres de compilació, i dins del fitxer **onos/modules.bzl** es defineix el OAR de la nova aplicació.

```
COMPILE_DEPS = [
    '//lib:CORE_DEPS',
]

osgi_jar_with_tests (
    deps = COMPILE_DEPS,
)

onos_app (
    title = 'Mastership Load Balancer',
    category = 'Utility',
    url = 'http://onosproject.org',
    description = 'Monitors distribution of mastership of network devices ' +
        'between the ONOS cluster nodes and periodically re-assigns ' +
        'the mastership to achieve balanced distribution when ' +
        'necessary and possible.',
)
```

**Fig 1.16** Exemple de fitxer BUCK.

```
osgi_jar_with_tests(  
    deps = CORE_DEPS,  
)  
  
onos_app(  
    category = "Utility",  
    description = "Monitors distribution of mastership of network devices " +  
                  "between the ONOS cluster nodes and periodically re-assigns " +  
                  "the mastership to achieve balanced distribution when " +  
                  "necessary and possible.",  
    title = "Mastership Load Balancer",  
    url = "http://onosproject.org",  
)
```

**Fig 1.17** Exemple de fitxer BUILD.

## CAPÍTOL 2. ESCENARIS BASATS EN ONOS

En aquest capítol s'expliquen dos escenaris d'ONOS basats en les versions 1.14 i 1.15 on s'han afegit dos aplicacions per controlar els dispositius que gestiona cada controlador, i també s'expliquen els fitxers que s'han de modificar per tal d'afegir-les.

La primera aplicació es diu **move-device**, serveix per canviar el controlador que gestiona una switch, i per executar-la s'ha de fer manualment des del CLI. Necessita que s'especifiquin dos paràmetres, la direcció MAC del switch que es pretén canviar el controlador de gestió, i la direcció IP del nou controlador que gestionarà el switch.

L'altra aplicació es diu **balancing** i serveix per moure un switch entre els controladors de la xarxa. Aquesta aplicació es pot executar de dos formes diferents, la primera és manualment a través del CLI, que mou el switch cada cop que s'executi per línia de comandes, i l'altra és que s'executi sola periòdicament cada X segons (aquest paràmetre X s'ha d'indicar dins la funció abans d'arrancar ONOS. Si es vol modificar aquest paràmetre s'ha de parar ONOS i tornar-lo a arrancar un cop s'hagi modificat).

Per cada escenari s'explica la topologia utilitzada amb les màquines utilitzades, i les comprovacions del seu funcionament.

### 2.1 Fitxers a modificar

Per afegir les dos aplicacions a ONOS s'han de modificar els fitxers que es descriuen a continuació (en aquest apartat només s'especifiquen els fitxers a modificar. Per veure el contingut que hi va a cadascun veure els annexos D i E):

- **MastershipAdminService.java** (es troba dins del directori `onos/core/api/src/main/java/org/onosproject/mastership/`)
- **MastershipManager.java** (es troba dins del directori `onos/core/net/src/main/java/org/onosproject/cluster/impl/`)
- **VirtualNetworkMastershipManager.java** (es troba dins del directori `onos/incubator/net/src/main/java/org/onosproject/incubator/net/virtual/impl/`).
- **shell-config.xml** (es troba dins del directori `onos/cli/src/main/resources/OSGI-INF/blueprint/`)
- Afegir una nova classe amb nom **MoveDeviceToNodeCommand** dins de `onos/cli/src/main/java/org/onosproject/cli/`

- Afegir una nova classe amb nom **BalancingCommand** dins de **onos/cli/src/main/java/org/onosproject/cli/**
- Dins del directori **onos/apps/** s'ha d'afegir un nou arbre de directoris: **/balancing/src/main/java/org/onosproject/balancing/** , i aquí dins s'han de crear dos fitxers nous, **Balancing.java** i **package-info.java**.
- Dins del directori **onos/apps/balancing/** crear un fitxer amb nom **BUCK**
- Dins del directori **onos/apps/balancing/** crear un fitxer amb nom **BUILD**
- **Modules.defs** (es troba dins del directori **onos/**)
- **Modules.bzl** (es troba dins del directori **onos/**)

## 2.2 Versió 1.14

Per instal·lar la versió 1.14 d'ONOS a les màquines veure l'annex A.

### 2.2.1 topologia utilitzada

Per aquest escenari s'utilitzen 3 màquines virtuals que es troben dins d'un servidor OpenStack de la UPC (adreça IP 147.83.118.244). El sistema operatiu de les tres màquines és Ubuntu 16.04.5 LTS i estan formades per 1VCPU, 2GB de memòria RAM i 30 GB de disc d'emmagatzematge. L'usuari que s'ha creat per fer les proves és **ubuntu**.

Per accedir a aquestes màquines s'ha de fer a través de ssh i l'autenticació es fa amb una clau ssh (key-onos.pem).

Com que la connexió es fa a través del terminal, es realitza un mapeig del port 8181 de cada màquina per accedir al GUI, a un port de la direcció IP pública del servidor OpenStack i així no tenir cap problema de connexió. Aquests ports es veuen a la següent taula, així com el nom de cada màquina, el port ssh per connectar-se i la seva direcció IP privada.

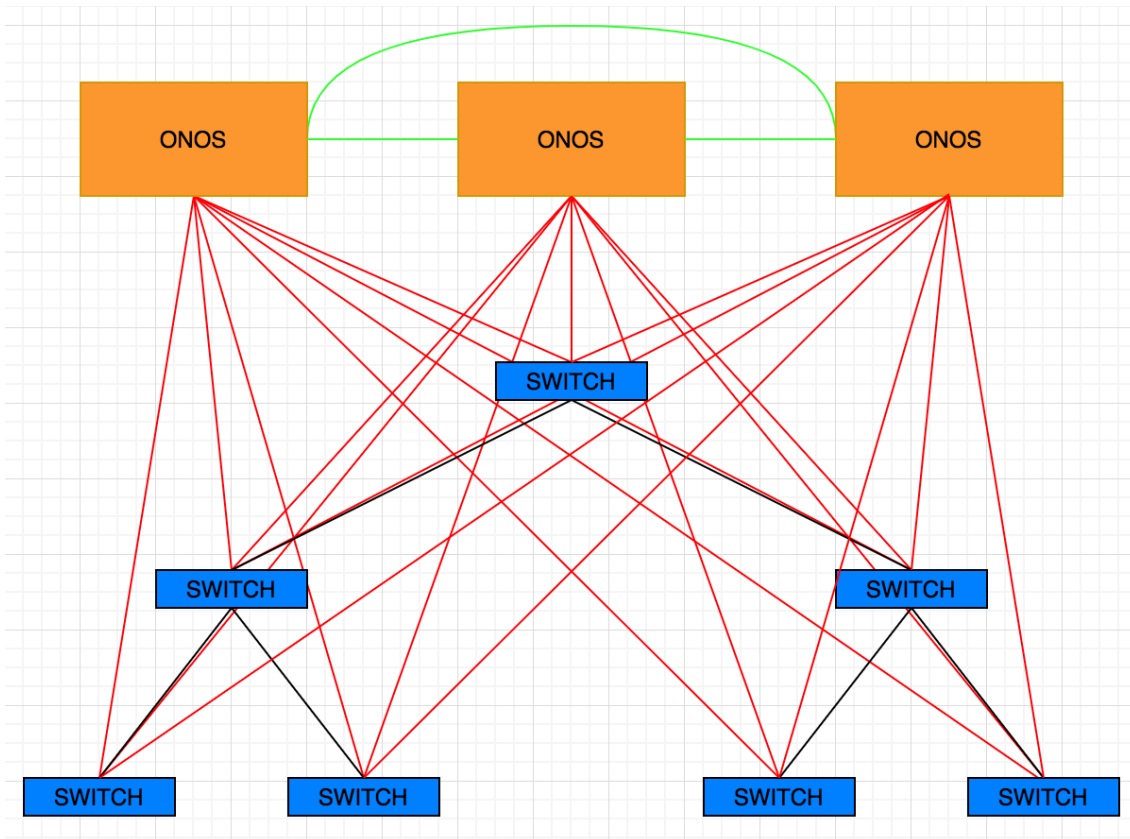
**Taula 2.1** Resum paràmetres màquines ONOS 1.14

Nom màquina	Port ssh	IP privada	Port GUI
Onos-9	128	192.168.60.23	8074
Onos-10	129	192.168.60.24	8075
Onos-11	132	192.168.60.25	8076

Per accedir a les diferents màquines via ssh s'han d'executar les següents comandes (és important executar-les des del directori on està guardada la clau ssh, sinó no s'estableix la connexió):

- Onos-9: `ssh -i key-onos.pem ubuntu@147.83.118.244 -Y -p 128`
- Onos-10: `ssh -i key-onos.pem ubuntu@147.83.118.244 -Y -p 129`
- Onos-11: `ssh -i key-onos.pem ubuntu@147.83.118.244 -Y -p 132`

Juntament amb els controladors hi ha 7 switches amb una topologia d'arbre tal i com es veu a la següent imatge:



**Fig 2.1** Topologia de xarxa de l'escenari d'ONOS 1.14.

A la figura anterior s'observa la topologia de xarxa mencionada amb els 3 controladors ONOS, 7 switches, i les connexions necessàries (de color verd els links entre controladors, de color negre els links entre switches i de color vermell els links entre controladors i switches).

### 2.2.2 Verificació del correcte funcionament de les noves aplicacions

Una vegada ja estiguin fetes totes les modificacions dels fitxers només falta compilar tot el codi, dins del directori de onos, amb la comanda **buck build onos**.

Quan hagi acabat de compilar el codi, executar onos amb la comanda **buck run onos-local -- clean**.

Arrancar onos a les tres màquines i un cop estiguin iniciades (esperar fins que no es generin més logs) entrar al CLI. **És molt important entrar al CLI de totes les màquines abans de crear el clúster perquè sinó després no deixa accedir-hi.**

Quan ja s'ha accedit al CLI de totes les màquines s'ha de crear el clúster. Per fer-ho, des d'alguna de les màquines, s'ha d'obrir un nou terminal i executar la comanda **./onos/tools/package/bin/onos-form-cluster -u karaf -p karaf 192.168.60.23 192.168.60.24 192.168.60.25**. Quan s'executi aquesta comanda, les tres instàncies actives d'ONOS es paren i tornen a arrancar amb la configuració del nou clúster. Esperar a que que tornin a estar iniciades com abans i tornar a accedir al CLI.

Quan ONOS ha acabat d'arrancar, entre els logs s'ha de veure la línia marcada que apareix a la següent imatge, el que significa que l'aplicació periòdica s'ha instal·lat correctament:

- 1.14.0.SNAPSHOT	Application org.onosproject.artemis has been installed
- 1.14.0.SNAPSHOT	Application org.onosproject.castor has been installed
- 1.14.0.SNAPSHOT	Application org.onosproject.balancing has been installed
- 1.14.0.SNAPSHOT	Application org.onosproject.drivers.ciena.waveserver.rest has b
- 1.14.0.SNAPSHOT	Application org.onosproject.intentperf has been installed
- 1.14.0.SNAPSHOT	Application org.onosproject.grpc.nb.service has been installed

**Fig 2.2** Logs d'ONOS on mostra que la nova aplicació s'ha instal·lat.

Per comprovar el funcionament de les noves aplicacions l'únic que falta és connectar algun dispositiu a la xarxa. Per fer-ho s'utilitzarà mininet<sup>2</sup>, i amb la següent comanda es crea la topologia d'arbre de 7 switches:

```
sudo mn --mac --controller=remote,ip=192.168.60.23 --  
controller=remote,ip=192.168.60.24 --controller=remote,ip=192.168.60.25 --  
topo=tree,3,2 --switch=ovsk,protocols=OpenFlow13
```

A la comanda de mininet se li ha d'especificar la direcció IP dels tres controladors, la topologia de xarxa, el tipus de switch i el protocol. A part d'això l'opció **--mac** serveix per reinicialitzar les direccions MAC dels dispositius.

Quan s'inicia la xarxa, s'assignen els switches de forma aleatòria entre els controladors que hi ha al clúster. Inicialment la repartició s'ha fet de la següent forma:

<sup>2</sup> Programari emulador per a la creació de prototips de xarxes en una sola màquina.

```
[onos> masters
192.168.60.23: 2 devices
  of:0000000000000002
  of:0000000000000004
192.168.60.24: 5 devices
  of:0000000000000001
  of:0000000000000003
  of:0000000000000005
  of:0000000000000006
  of:0000000000000007
192.168.60.25: 0 devices
onos>
```

**Fig 2.3** Repartició dels switches entre els controladors.

A continuació s'ha de provar que l'aplicació **move-device** funciona correctament. Per fer-ho, per exemple, es mouen els switches of:0000000000000003 i of:0000000000000007 cap al controlador 192.168.60.25.

```
[onos> masters
192.168.60.23: 2 devices
  of:0000000000000002
  of:0000000000000004
192.168.60.24: 5 devices
  of:0000000000000001
  of:0000000000000003
  of:0000000000000005
  of:0000000000000006
  of:0000000000000007
192.168.60.25: 0 devices
[nos> move-device of:0000000000000003 192.168.60.25
[nos> move-device of:0000000000000007 192.168.60.25
[nos> masters
192.168.60.23: 2 devices
  of:0000000000000002
  of:0000000000000004
192.168.60.24: 3 devices
  of:0000000000000001
  of:0000000000000005
  of:0000000000000006
192.168.60.25: 2 devices
  of:0000000000000003
  of:0000000000000007
onos> █
```

**Fig 2.4** Repartició dels switches entre els controladors abans i després d'utilitzar la nova aplicació.

Com s'observa a la figura anterior, inicialment els switches of:0000000000000003 i of:0000000000000007 estan gestionats pel controlador 192.168.60.24 i després d'executar la comanda de la nova aplicació especificant

els switches i el controlador de destí, els dos switches passen a estar gestionats pel controlador 192.168.60.25.

Per provar l'aplicació **balancing** executant-la manualment des del CLI, abans de provar-la s'ha d'executar la comanda **balance-masters** per tal que els switches estiguin repartits d'una forma equitativa entre els controladors. A continuació s'executa la comanda **balancing** per moure el switch de controlador alternat amb la comanda **masters** per veure l'ocupació dels switches abans i després que el switch canviï de controlador.

```

onos> masters
192.168.60.23: 4 devices
  of:0000000000000003
  of:0000000000000004
  of:0000000000000005
  of:0000000000000006
192.168.60.24: 1 devices
  of:0000000000000002
192.168.60.25: 2 devices
  of:0000000000000001
  of:0000000000000007
onos> balancing
onos> masters
192.168.60.23: 3 devices
  of:0000000000000004
  of:0000000000000005
  of:0000000000000006
192.168.60.24: 1 devices
  of:0000000000000002
192.168.60.25: 3 devices
  of:0000000000000001
  of:0000000000000003
  of:0000000000000007
onos> balancing
onos> masters
192.168.60.23: 3 devices
  of:0000000000000004
  of:0000000000000005
  of:0000000000000006
192.168.60.24: 2 devices
  of:0000000000000002
  of:0000000000000003
192.168.60.25: 2 devices
  of:0000000000000001
  of:0000000000000007
onos> balancing
onos> masters
192.168.60.23: 4 devices
  of:0000000000000003
  of:0000000000000004
  of:0000000000000005
  of:0000000000000006
192.168.60.24: 1 devices
  of:0000000000000002
192.168.60.25: 2 devices
  of:0000000000000001
  of:0000000000000007

```

**Fig 2.5** Repartició dels switches entre els controladors abans, durant i després d'utilitzar la nova aplicació.

Com es pot observar a la figura anterior, inicialment el switch of:0000000000000003 està gestionat pel controlador 192.168.60.23 i després d'executar la comanda de la nova aplicació per primer cop, ha passat a estar gestionat pel controlador 192.168.60.25. Posteriorment, després d'executar la comanda per segon cop, ha passat a estar gestionat pel controlador



192.168.60.24, i en l'última execució ha tornat a estar gestionat pel controlador 192.168.60.23.

Per provar l'aplicació periòdica, igual que s'ha fet quan s'ha provat l'aplicació pel CLI, abans de provar-la, s'ha d'executar la comanda **balance-masters** per tal que els switches estiguin repartits d'una forma equitativa entre els controladors.

A continuació s'ha de provar que la nova aplicació funciona correctament. Per fer-ho, s'activa mitjançant la comanda **app activate balancing** des de qualsevol dels controladors. Això activa l'aplicació i informa a la resta de nodes d'activar-la també.

```
[onos> app activate balancing
Activated org.onosproject.balancing
[onos> masters
192.168.60.23: 3 devices
  of:0000000000000003
  of:0000000000000004
  of:0000000000000007
192.168.60.24: 2 devices
  of:0000000000000005
  of:0000000000000006
192.168.60.25: 2 devices
  of:0000000000000001
  of:0000000000000002
[onos> masters
192.168.60.23: 2 devices
  of:0000000000000004
  of:0000000000000007
192.168.60.24: 2 devices
  of:0000000000000005
  of:0000000000000006
192.168.60.25: 3 devices
  of:0000000000000001
  of:0000000000000002
  of:0000000000000003
[onos> masters
192.168.60.23: 2 devices
  of:0000000000000004
  of:0000000000000007
192.168.60.24: 3 devices
  of:0000000000000003
  of:0000000000000005
  of:0000000000000006
192.168.60.25: 2 devices
  of:0000000000000001
  of:0000000000000002
```

**Fig 2.6** Repartició dels switches entre els controladors abans, durant i després d'utilitzar la nova aplicació periòdica.

```

2018-10-18 07:50:43,898 | INFO | Balancing-0 | Balancing | 177 - org.onosproject.onos
-apps-balancing - 1.14.0.SNAPSHOT |
balancing app

2018-10-18 07:50:43,910 | INFO | Balancing-0 | MastershipManager | 130 - org.onosproject.onos
-core-net - 1.14.0.SNAPSHOT | Setting 192.168.60.25 as the master for of:0000000000000003
2018-10-18 07:50:43,918 | INFO | Balancing-0 | Balancing | 177 - org.onosproject.onos
-apps-balancing - 1.14.0.SNAPSHOT | Completed balance roles
2018-10-18 07:50:46,923 | INFO | onos-of-worker-0 | ntrollerImpl$OpenFlowSwitchAgent | 171 - org.onosproject.onos
-protocols-openflow-ctl - 1.14.0.SNAPSHOT | Transitioned switch 00:00:00:00:00:00:03 to EQUAL
2018-10-18 07:51:13,923 | INFO | Balancing-0 | Balancing | 177 - org.onosproject.onos
-apps-balancing - 1.14.0.SNAPSHOT |
balancing app

2018-10-18 07:51:13,929 | INFO | Balancing-0 | MastershipManager | 130 - org.onosproject.onos
-core-net - 1.14.0.SNAPSHOT | Setting 192.168.60.25 as the master for of:0000000000000003
2018-10-18 07:51:13,931 | INFO | Balancing-0 | Balancing | 177 - org.onosproject.onos
-apps-balancing - 1.14.0.SNAPSHOT | Completed balance roles
2018-10-18 07:51:43,932 | INFO | Balancing-0 | Balancing | 177 - org.onosproject.onos
-apps-balancing - 1.14.0.SNAPSHOT |
balancing app

2018-10-18 07:51:43,941 | INFO | Balancing-0 | MastershipManager | 130 - org.onosproject.onos
-core-net - 1.14.0.SNAPSHOT | Setting 192.168.60.23 as the master for of:0000000000000003
2018-10-18 07:51:43,944 | INFO | Balancing-0 | Balancing | 177 - org.onosproject.onos
-apps-balancing - 1.14.0.SNAPSHOT | Completed balance roles
2018-10-18 07:51:46,952 | INFO | onos-of-worker-0 | ntrollerImpl$OpenFlowSwitchAgent | 171 - org.onosproject.onos
-protocols-openflow-ctl - 1.14.0.SNAPSHOT | Transitioned switch 00:00:00:00:00:00:03 to MASTER
2018-10-18 07:52:13,946 | INFO | Balancing-0 | Balancing | 177 - org.onosproject.onos
-apps-balancing - 1.14.0.SNAPSHOT |
balancing app

2018-10-18 07:52:13,954 | INFO | Balancing-0 | MastershipManager | 130 - org.onosproject.onos
-core-net - 1.14.0.SNAPSHOT | Setting 192.168.60.25 as the master for of:0000000000000003
2018-10-18 07:52:13,958 | INFO | Balancing-0 | Balancing | 177 - org.onosproject.onos
-apps-balancing - 1.14.0.SNAPSHOT | Completed balance roles
2018-10-18 07:52:16,967 | INFO | onos-of-worker-0 | ntrollerImpl$OpenFlowSwitchAgent | 171 - org.onosproject.onos
-protocols-openflow-ctl - 1.14.0.SNAPSHOT | Transitioned switch 00:00:00:00:00:00:03 to EQUAL

```

**Fig 2.7** Funcionament aplicació balancing vist des dels logs d'ONOS.

Com s'observa ar a les dos figures anteriors, primer s'activa l'aplicació i a partir d'aquest moment, cada 30 segons (paràmetre configurat dins del fitxer **Balancing.java**) s'executa la funció de balancing i es veu com el switch of:0000000000000003 canvia de controlador.

## 2.3 Versió 1.15

Per instal·lar la versió 1.15 d'ONOS a les màquines veure l'annex A.

### 2.3.1 topologia utilitzada

En aquest escenari s'utilitzen sis màquines virtuals (tres d'elles amb ONOS instal·lat i les altres tres amb la versió 3.0.5 d'Atomix instal·lat (instal·lació explicada a l'annex B)) que es troben dins d'un servidor OpenStack de la UPC (adreça IP 147.83.118.244). El sistema operatiu de les sis màquines és Ubuntu 16.04.5 LTS.

Les tres màquines d'ONOS estan formades per 1VCPU, 2GB de memòria RAM i 30 GB de disc d'emmagatzematge, i les tres màquines d'Atomix estan formades per 1VCPU, 2GB de memòria RAM i 20 GB de disc d'emmagatzematge. L'usuari que s'ha creat per fer les proves és **ubuntu**.

Per accedir a aquestes màquines s'ha de fer a través de ssh i l'autenticació es fa amb una clau ssh (key-onos.pem).

Com que la connexió es fa a través del terminal, es realitza un mapeig del port 8181 de cada màquina per accedir al GUI, a un port de la direcció IP pública del servidor OpenStack i així no tenir cap problema de connexió. Aquests ports es veuen a la següent taula, així com el nom de cada màquina, el port ssh per connectar-se i la seva direcció IP privada.

**Taula 2.2** Resum paràmetres màquines ONOS 1.15

Nom màquina	Port ssh	IP privada	Port GUI
Onos-1	106	192.168.60.14	8092
Onos-2	107	192.168.60.22	8093
Onos-5	118	192.168.60.13	8094

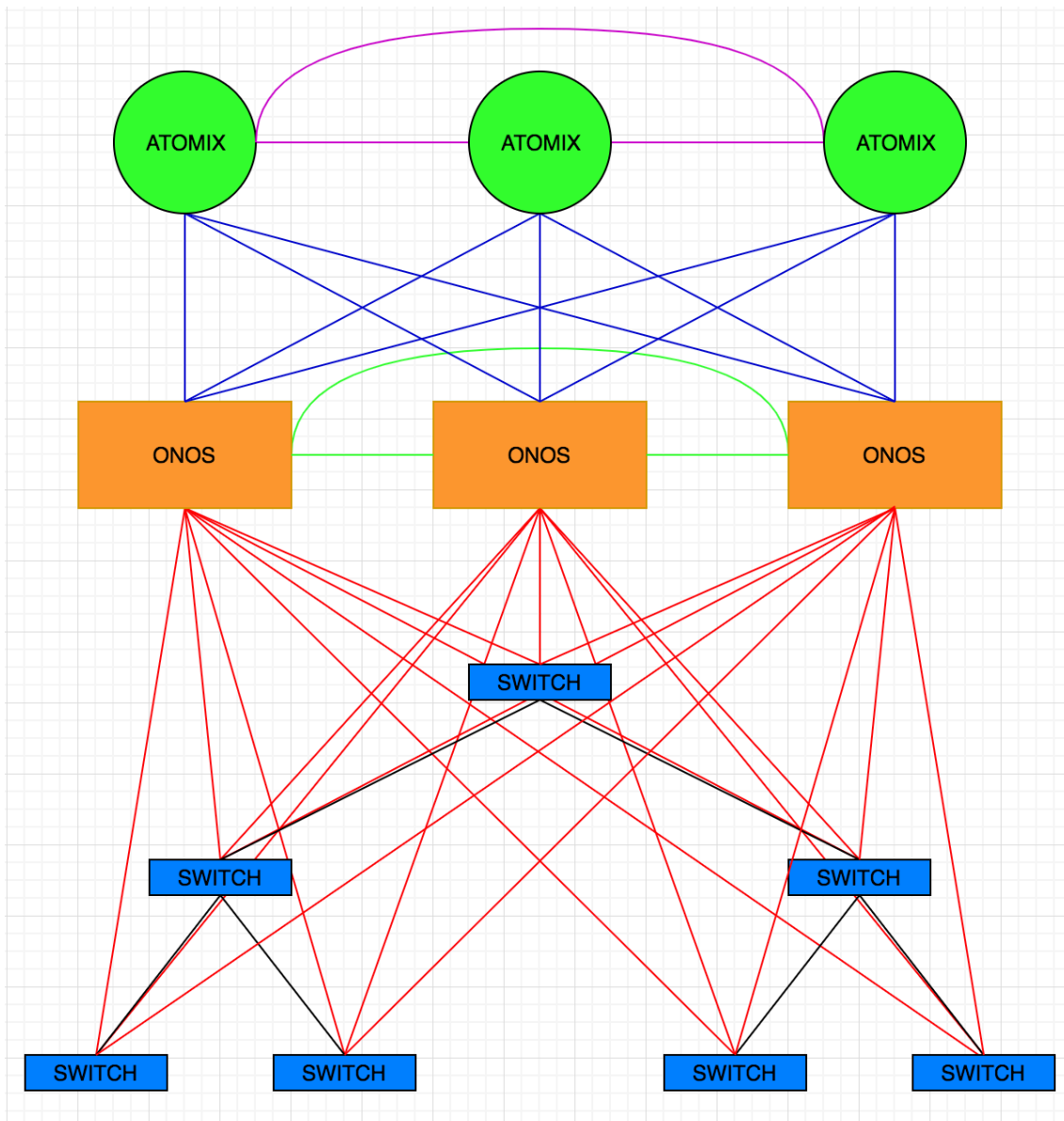
**Taula 2.3** Resum paràmetres màquines Atomix 3.0.5

Nom màquina	Port ssh	IP privada
Onos-6	112	192.168.60.16
Onos-7	123	192.168.60.17
Onos-8	124	192.168.60.18

Per accedir a les diferents màquines via ssh s'han d'executar les següents comandes (és important executar-les des del directori on està guardada la clau ssh, sinó no s'estableix la connexió):

- Onos-1: **ssh -i key-onos.pem ubuntu@147.83.118.244 -Y -p 106**
- Onos-2: **ssh -i key-onos.pem ubuntu@147.83.118.244 -Y -p 107**
- Onos-5: **ssh -i key-onos.pem ubuntu@147.83.118.244 -Y -p 118**
- Onos-6: **ssh -i key-onos.pem ubuntu@147.83.118.244 -Y -p 112**
- Onos-7: **ssh -i key-onos.pem ubuntu@147.83.118.244 -Y -p 123**
- Onos-8: **ssh -i key-onos.pem ubuntu@147.83.118.244 -Y -p 124**

Juntament amb els controladors i els nodes d'Atomix hi ha 7 switches amb una topologia d'arbre tal i com es veu a la següent imatge:



**Fig 2.8** Topologia de xarxa de l'escenari d'ONOS 1.15.

A la figura anterior s'observa la topologia de xarxa mencionada amb els 3 controladors ONOS i els 3 nodes Atomix, 7 switches, i les connexions necessàries (de color verd els links entre controladors, de color negre els links entre switches, de color lila els links entre els nodes Atomix, de color blau els links entre els controladors i els nodes Atomix i de color vermell els links entre controladors i switches).

### 2.3.2 Configuracions addicionals per la versió 1.15

Per aquesta nova versió d'ONOS hi ha certes variacions respecte la versió 1.14. La primera variació és que ja no s'utilitza **Buck** per tal de compilar i executar el codi sinó que ara s'utilitza **Bazel**. Per poder utilitzar Buck no era necessari

instal·lar res, es podia executar directament. En canvi **Bazel** s'ha d'instal·lar prèviament per poder utilitzar-lo. Els passos a seguir per la seva instal·lació estan descrits a l'annex C. Un cop estigui instal·lat ja es pot utilitzar fàcilment.

L'altre canvi respecte la versió 1.14 és que la gestió del clúster ja no la fa ONOS, sinó que ara la fan uns nous nodes anomenats Atomix (explicat al punt 1.3.3). Per tant primer s'han d'instal·lar aquests nodes, procés explicat a l'annex B, i configurar-los. L'única configuració que necessiten un cop instal·lats es troba al fitxer `atomix.conf` (situat dins de **atomix/conf/**), i s'hi especifica el següent:

```
cluster {
  cluster-id: onos
  node.id: atomix-1
  node.address: "192.168.60.16:5679"
  discovery {
    type: bootstrap
    nodes.1 {
      id: atomix-1
      address: "192.168.60.16:5679"
    }
    nodes.2 {
      id: atomix-2
      address: "192.168.60.17:5679"
    }
    nodes.3 {
      id: atomix-3
      address: "192.168.60.18:5679"
    }
  }
}

management-group {
  type: raft
  partitions: 1
  storage.level: disk
  members: [atomix-1, atomix-2, atomix-3]
}

partition-groups.raft {
  type: raft
  partitions: 7
  storage.level: disk
  members: [atomix-1, atomix-2, atomix-3]
}
```

**Fig 2.9** Fitxer de configuració `atomix.conf` del node `atomix-1`.

En aquest fitxer primer s'hi d'especifica el **cluster-id** (nom desitjat pel clúster), el **node.id** (nom desitjat pel node), i el **node.address** (s'hi especifica la direcció IP privada de la màquina). A continuació s'hi introdueix tots els nodes que formaran el clúster d'Atomix indicant el **id** i la **direcció IP** de tots els nodes (inclòs el node local). Per últim s'hi defineix també el **management-group** i el **partitions-groups.raft** (aquesta configuració és sempre igual per tots els nodes). Les tres màquines d'Atomix han de tenir aquest fitxer de configuració, i l'única part que varia entre ells és la primera part on es defineix el **node.id** i el **node.address**.

Quan les tres màquines tenen la configuració que els hi toca s'inicia el servei. Per fer-ho s'ha d'introduir la següent comanda (exemple pel node `atomix-1`):

**bin/atomix-agent -m atomix-1 -a 10.192.19.112:5679**. L'opció **-m** és per indicar el node.id i l'opció **-a** és per indicar el node.address seguit del port 5679 (port per rebre les connexions).

Un cop introduïda la comanda anterior s'ha d'esperar fins que es connecten entre ells i es veu el següent al terminal:

```
10:08:15.323 [raft-partition-group-raft-1] INFO i.a.p.r.partition.RaftPartitionGroup - Started
10:08:15.323 [raft-partition-group-raft-1] INFO i.a.p.p.impl.DefaultPartitionService - Started
10:08:15.462 [raft-partition-group-system-2] INFO i.a.core.impl.CoreTransactionService - Started
10:08:15.462 [raft-partition-group-system-2] INFO i.a.core.impl.CorePrimitivesService - Started
10:08:15.465 [main] INFO io.atomix.agent.AtomixAgent - Atomix listening at 192.168.60.16:5679
10:08:18.175 [vert.x-eventloop-thread-1] INFO io.atomix.rest.impl.VertxRestService - Started
10:08:18.177 [main] INFO io.atomix.agent.AtomixAgent - HTTP server listening at 192.168.60.16:5678
```

**Fig 2.10** Logs d'Atomix del node atomix-1.

Quan apareixen aquestes línies als logs ja es poden arrancar els nodes d'ONOS per crear el clúster. Però abans de fer-ho falta una última modificació per tal que el clúster es creï correctament. Es tracta d'indicar als nodes d'ONOS on estan ubicats els nodes Atomix atenent les peticions per unir-se al clúster.

Per fer-ho s'ha de modificar el fitxer **onos-run-karaf** (situat a **onos/tools/package**) i a la línia 56, on crea un fitxer cluster.json s'ha de posar la següent configuració:

```
# Create config/cluster.json (cluster metadata)
IP=${ONOS_IP:-192.168.60.13}
echo "Creating local cluster configs for IP $IP..."
[ -d $ONOS_DIR/config ] || mkdir -p $ONOS_DIR/config
cat > $ONOS_DIR/config/cluster.json <<-EOF
{
  "node": {
    "ip": "192.168.60.13",
    "id": "192.168.60.13",
    "port": 9876
  },
  "storage": [
    {
      "ip": "192.168.60.16",
      "id": "atomix-1",
      "port": 5679
    },
    {
      "ip": "192.168.60.17",
      "id": "atomix-2",
      "port": 5679
    },
    {
      "ip": "192.168.60.18",
      "id": "atomix-3",
      "port": 5679
    }
  ],
  "name": "onos"
}
EOF
```

**Fig 2.11** Fitxer de configuració onos-run-karaf.

### 2.3.3 Verificació del correcte funcionament de les noves aplicacions

Quan ja estiguin fetes totes les modificacions i els nodes d'Atomix en funcionament s'ha de compilar tot el codi amb la comanda **bazel build onos**. Quan hagi acabat de compilar el codi, executar onos amb la comanda **bazel run onos-local -- clean**.

Arrancar onos a les tres màquines de la mateixa forma i un cop estan iniciades (esperar fins que no es generen més logs) entrar al CLI. En aquesta versió el clúster s'inicia automàticament, per tant no cal executar cap comanda específica per tal de crear-lo.

Quan ONOS ha acabat d'arrancar, entre els logs s'ha de veure la línia marcada que apareix a la següent imatge, el que significa que l'aplicació periòdica s'ha instal·lat correctament:

- 1.15.0.SNAPSHOT	Started
- 1.15.0.SNAPSHOT	Started
- 1.15.0.SNAPSHOT	Application org.onosproject.balancing has been installed
- 1.15.0.SNAPSHOT	Application org.onosproject.castor has been installed
- 1.15.0.SNAPSHOT	Started

**Fig 2.12** Logs d'ONOS on mostra la nova aplicació instal·lada.

Ara, per comprovar el funcionament de la nova aplicació l'únic que falta és connectar els dispositius a la xarxa. Per fer-ho s'utilitza la mateixa xarxa de mininet que s'ha utilitzat anteriorment amb la comanda:

```
sudo mn --mac --controller=remote,ip=192.168.60.23 --  
controller=remote,ip=192.168.60.24 --controller=remote,ip=192.168.60.25 --  
topo=tree,3,2 --switch=ovsk,protocols=OpenFlow13
```

Quan s'inicia la xarxa, s'assignen els switches de forma aleatòria entre els controladors que hi ha al clúster. Inicialment la repartició s'ha fet de la següent forma:

```
[onos> masters
192.168.60.13: 1 devices
  of:0000000000000005
192.168.60.14: 3 devices
  of:0000000000000001
  of:0000000000000003
  of:0000000000000006
192.168.60.22: 3 devices
  of:0000000000000002
  of:0000000000000004
  of:0000000000000007
onos> █
```

**Fig 2.13** Repartició dels switches entre els controladors.

A continuació, per provar el correcte funcionament de l'aplicació **move-device**, es mou el switch of:000000000000000001 al controlador 192.168.60.13 i el switch of:000000000000000004 al controlador 192.168.60.14

```
[onos> masters
192.168.60.13: 1 devices
  of:000000000000000005
192.168.60.14: 3 devices
  of:000000000000000001
  of:000000000000000003
  of:000000000000000006
192.168.60.22: 3 devices
  of:000000000000000002
  of:000000000000000004
  of:000000000000000007
[onos> move-device of:000000000000000001 192.168.60.13
[onos> move-device of:000000000000000004 192.168.60.14
[onos> masters
192.168.60.13: 2 devices
  of:000000000000000001
  of:000000000000000005
192.168.60.14: 3 devices
  of:000000000000000003
  of:000000000000000004
  of:000000000000000006
192.168.60.22: 2 devices
  of:000000000000000002
  of:000000000000000007
onos> █
```

**Fig 2.14** Repartició dels switches entre els controladors abans i després d'utilitzar la nova aplicació.

Com s'observa a la figura anterior, inicialment el switch of:000000000000000001 està gestionat pel controlador 192.168.60.14 i el switch of:000000000000000004 està gestionat pel controlador 192.168.60.22. Després d'utilitzar la nova aplicació els switches passen a estar gestionats pels controladors 192.168.60.13 i 192.168.60.14 respectivament.

Per provar l'aplicació **balancing** executant-la manualment des del CLI, abans de provar-la s'ha d'executar la comanda **balance-masters** per tal que els switches estiguin repartits d'una forma equitativa entre els controladors. A continuació s'executa la comanda **balancing** per moure el switch de controlador alternat amb la comanda **masters** per veure l'ocupació dels switches abans i després que el switch canviï de controlador.



```

onos> masters
192.168.60.13: 2 devices
  of:0000000000000003
  of:0000000000000005
192.168.60.14: 3 devices
  of:0000000000000002
  of:0000000000000006
  of:0000000000000007
192.168.60.22: 2 devices
  of:0000000000000001
  of:0000000000000004
onos> balancing
onos> masters
192.168.60.13: 1 devices
  of:0000000000000005
192.168.60.14: 3 devices
  of:0000000000000002
  of:0000000000000006
  of:0000000000000007
192.168.60.22: 3 devices
  of:0000000000000001
  of:0000000000000003
  of:0000000000000004
onos> balancing
onos> masters
192.168.60.13: 1 devices
  of:0000000000000005
192.168.60.14: 4 devices
  of:0000000000000002
  of:0000000000000003
  of:0000000000000006
  of:0000000000000007
192.168.60.22: 2 devices
  of:0000000000000001
  of:0000000000000004
onos> balancing
onos> masters
192.168.60.13: 2 devices
  of:0000000000000003
  of:0000000000000005
192.168.60.14: 3 devices
  of:0000000000000002
  of:0000000000000006
  of:0000000000000007
192.168.60.22: 2 devices
  of:0000000000000001
  of:0000000000000004
onos>

```

**Fig 2.15** Repartició dels switches entre els controladors abans, durant i després d'utilitzar la nova aplicació.

Com s'observa a la figura anterior, inicialment el switch of:0000000000000003 està gestionat pel controlador 192.168.60.23 i després d'executar la comanda de la nova aplicació per primer cop, ha passat a estar gestionat pel controlador 192.168.60.25. Posteriorment, després d'executar la comanda per segon cop, ha passat a estar gestionat pel controlador 192.168.60.24, i en l'última execució ha tornat a estar gestionat pel controlador 192.168.60.23.

Per provar l'aplicació periòdica, igual que s'ha fet quan s'ha provat l'aplicació pel CLI, abans de provar-la, s'ha d'executar la comanda **balance-masters** per tal que els switches estiguin repartits d'una forma equitativa entre els controladors.

A continuació s'ha de provar que la nova aplicació funciona correctament. Per fer-ho, s'activa mitjançant la comanda **app activate balancing** des de qualsevol

dels controladors. Això activa l'aplicació i informa a la resta de nodes d'activar-la també.

```
[onos> app activate balancing
Activated org.onosproject.balancing
[onos> masters
192.168.60.13: 3 devices
  of:000000000000000003
  of:000000000000000005
  of:000000000000000006
192.168.60.14: 2 devices
  of:000000000000000001
  of:000000000000000002
192.168.60.22: 2 devices
  of:000000000000000004
  of:000000000000000007
[onos> masters
192.168.60.13: 2 devices
  of:000000000000000005
  of:000000000000000006
192.168.60.14: 2 devices
  of:000000000000000001
  of:000000000000000002
192.168.60.22: 3 devices
  of:000000000000000003
  of:000000000000000004
  of:000000000000000007
[onos> masters
192.168.60.13: 2 devices
  of:000000000000000005
  of:000000000000000006
192.168.60.14: 3 devices
  of:000000000000000001
  of:000000000000000002
  of:000000000000000003
192.168.60.22: 2 devices
  of:000000000000000004
  of:000000000000000007
[onos> masters
192.168.60.13: 3 devices
  of:000000000000000003
  of:000000000000000005
  of:000000000000000006
192.168.60.14: 2 devices
  of:000000000000000001
  of:000000000000000002
192.168.60.22: 2 devices
  of:000000000000000004
  of:000000000000000007
onos> █
```

**Fig 2.16** Repartició dels switches entre els controladors abans, durant i després d'activar la nova aplicació.

```

2018-10-19 11:10:46,728 | INFO | Balancing-0 | Balancing | 191 - org.onosproject.onos-
apps-balancing - 1.15.0.SNAPSHOT | balancing app
2018-10-19 11:10:46,740 | INFO | Balancing-0 | MastershipManager | 161 - org.onosproject.onos-
core-net - 1.15.0.SNAPSHOT | Setting 192.168.60.22 as the master for of:0000000000000003
2018-10-19 11:10:46,751 | INFO | Balancing-0 | Balancing | 191 - org.onosproject.onos-
apps-balancing - 1.15.0.SNAPSHOT | Completed balance roles
2018-10-19 11:10:49,764 | INFO | onos-of-worker-0 | ntrollerImpl$OpenFlowSwitchAgent | 178 - org.onosproject.onos-
protocols-openflow-ctl - 1.15.0.SNAPSHOT | Transitioned switch 00:00:00:00:00:00:03 to EQUAL

2018-10-19 11:11:06,751 | INFO | Balancing-0 | Balancing | 191 - org.onosproject.onos-
apps-balancing - 1.15.0.SNAPSHOT | balancing app
2018-10-19 11:11:06,758 | INFO | Balancing-0 | MastershipManager | 161 - org.onosproject.onos-
core-net - 1.15.0.SNAPSHOT | Setting 192.168.60.14 as the master for of:0000000000000003
2018-10-19 11:11:06,767 | INFO | Balancing-0 | Balancing | 191 - org.onosproject.onos-
apps-balancing - 1.15.0.SNAPSHOT | Completed balance roles

2018-10-19 11:11:26,769 | INFO | Balancing-0 | Balancing | 191 - org.onosproject.onos-
apps-balancing - 1.15.0.SNAPSHOT | balancing app
2018-10-19 11:11:26,775 | INFO | Balancing-0 | MastershipManager | 161 - org.onosproject.onos-
core-net - 1.15.0.SNAPSHOT | Setting 192.168.60.13 as the master for of:0000000000000003
2018-10-19 11:11:26,785 | INFO | Balancing-0 | Balancing | 191 - org.onosproject.onos-
apps-balancing - 1.15.0.SNAPSHOT | Completed balance roles

2018-10-19 11:11:29,799 | INFO | ow-backup-sender | DeviceFlowTable | 163 - org.onosproject.onos-
core-dist - 1.15.0.SNAPSHOT | Synchronizing device of:0000000000000003 flows for term 34
2018-10-19 11:11:29,800 | INFO | onos-of-worker-0 | ntrollerImpl$OpenFlowSwitchAgent | 178 - org.onosproject.onos-
protocols-openflow-ctl - 1.15.0.SNAPSHOT | Transitioned switch 00:00:00:00:00:00:03 to MASTER

2018-10-19 11:11:46,787 | INFO | Balancing-0 | Balancing | 191 - org.onosproject.onos-
apps-balancing - 1.15.0.SNAPSHOT | balancing app
2018-10-19 11:11:46,796 | INFO | Balancing-0 | MastershipManager | 161 - org.onosproject.onos-
core-net - 1.15.0.SNAPSHOT | Setting 192.168.60.22 as the master for of:0000000000000003
2018-10-19 11:11:46,807 | INFO | Balancing-0 | Balancing | 191 - org.onosproject.onos-
apps-balancing - 1.15.0.SNAPSHOT | Completed balance roles
2018-10-19 11:11:49,821 | INFO | onos-of-worker-0 | ntrollerImpl$OpenFlowSwitchAgent | 178 - org.onosproject.onos-
protocols-openflow-ctl - 1.15.0.SNAPSHOT | Transitioned switch 00:00:00:00:00:00:03 to EQUAL
2018-10-19 11:11:53,804 | INFO | p-app-activation | FeaturesServiceImpl | 20 - org.apache.karaf.featu
res.core - 3.0.8 | Uninstalling feature onos-apps-balancing 1.15.0-SNAPSHOT
2018-10-19 11:11:53,812 | INFO | p-app-activation | Balancing | 191 - org.onosproject.onos-
apps-balancing - 1.15.0.SNAPSHOT | Stopped
2018-10-19 11:11:53,821 | INFO | p-app-activation | ApplicationManager | 161 - org.onosproject.onos-

```

**Fig 2.17** Funcionament aplicació balancing vist des dels logs d'ONOS.

Com s'observa a les dos figures anteriors, primerament s'activa l'aplicació i a partir d'aquest moment, cada 20 segons (paràmetre configurat dins del fitxer **Balancing.java**) s'executa la funció de balancing i es veu com el switch of:0000000000000003 canvia de controlador.

## CAPÍTOL 3. CONCLUSIONS

Un cop finalitzat aquest projecte es pot concloure que l'aparició de tecnologies com SDN aporten un gran nombre d'avantatges pel desplegament de noves xarxes ja que hi ha un gran estalvi econòmic amb l'aparició de dispositius no propietaris, i per la fàcil integració i creació de serveis virtuals.

Malgrat ser un camp encara en investigació i desenvolupament pel gran ventall d'oportunitats que pot oferir i que encara no s'han explorat, ja hi ha varies implementacions desplegades en xarxes reals.

En concret pel projecte ONOS, tot i fer tot just 4 anys que està en desenvolupament, actualment ja van per la versió 1.15 amb el suport d'un gran nombre de contribuïdors que fan possible la ràpida evolució d'aquest controlador de xarxa.

En aquest projecte s'han utilitzat les versions 1.14 i 1.15 i s'han provat totes les diferències que hi ha entre les dos versions. La diferència més gran és respecte a la formació del clúster.

A la versió 1.14 el clúster s'havia de crear de forma manual a través d'un script al qual se li ha d'especificar les direccions IP dels controladors que formaran part del clúster, i un cop creat no es pot unir cap controlador més (a no ser que es destrueixi el clúster i se'n creï un de nou especificant també el nou controlador) ni si un controlador es desconnecta pot tornar a unir-se.

En canvi a la versió 1.15 s'ha externalitzat el servei del clúster i el passa a gestionar un conjunt de nodes Atomix que atenen les peticions de connexió o desconnexió al clúster dels controladors ONOS. Per fer-ho s'han de configurar els nodes Atomix paral·lelament a ONOS, i als nodes d'ONOS se'ls ha d'especificar les direccions on es troben els nodes Atomix per poder comunicar-se amb ells.

Posteriorment s'han implementat satisfactòriament dos aplicacions per tal de gestionar els dispositius connectats als controladors de la xarxa. Aquestes aplicacions són **move-device** i **balancing**.

La primera permet moure manualment la gestió d'un switch en concret d'un controlador cap a un altre, i la segona permet moure tant manualment com automàticament (cada X segons) un switch entre els controladors de la xarxa.

La gran varietat de vies d'investigació que actualment té ONOS permet definir varies línies futures per ampliar el que s'ha realitzat en aquest projecte així com explorar altres vies que proporciona ONOS i no s'han estudiat durant la realització d'aquest projecte. Algunes possibles línies futures són:

- Implementar un algoritme de coordinació (dins de l'aplicació creada en aquest projecte) que faci una assignació dels switches en funció de l'estat de la xarxa tenint en compte paràmetres com l'estat dels links, pèrdues de paquets, ubicació física dels dispositius per disminuir retards, etc.

- Implementar algoritmes de coordinació en funció de les necessitats que no tinguin relació amb el punt anterior.
- Fer un desplegament d'ONOS en xarxes òptiques (de cara a la utilització d'ONOS en xarxes basades en fibra òptica).

## Bibliografia

- [1] Galeano, S., “El número de usuarios de Internet en el mundo supera el 50% de la población: 4.000 millones (2018)”, *marketing4ecommerce*.  
<https://marketing4ecommerce.net/usuarios-internet-mundo-2017/>
- [2] Definició SDN. <https://www.opennetworking.org/sdn-definition/>
- [3] ONOS project. <https://onosproject.org/>
- [4] Pàgina oficial d'Atomix. <https://atomix.io/>
- [5] Pàgina oficial de Bazel. <https://bazel.build/>
- [6] Open Network Foundation (ONF). <https://www.opennetworking.org/>
- [7] CLI d'ONOS. <https://wiki.onosproject.org/display/ONOS/The+ONOS+CLI>
- [8] GUI d'ONOS.  
<https://wiki.onosproject.org/display/ONOS/The+ONOS+Web+GUI>
- [9] Configuració clúster versió vella.  
<https://wiki.onosproject.org/display/ONOS/Forming+a+cluster>
- [10] Configuració clúster nova versió.  
<https://wiki.onosproject.org/pages/viewpage.action?pageId=28836788>
- [11] Fòrum de discussió d'ONOS.  
<https://groups.google.com/a/onosproject.org/forum/#!/forum/onos-discuss>
- [12] Fòrum de desenvolupament d'ONOS.  
<https://groups.google.com/a/onosproject.org/forum/#!/forum/onos-dev>

## Acrònims

API	Aplication Programing Interface
CLI	Command Line Interface
GUI	Graphical User Interface
IoT	Internet of Things
IP	Internet Protocol
IPTV	Internet Protocol Television
MAC	Media Access Control
OAR	ONOS Application aRchive
ONF	Open Network Foundation
ONOS	Open Network Operating System
RAM	Random Access Memory
SDN	Software Defined Networks
SSH	Secure Shell
VCPU	Virtual Central Processing Unit
VoD	Video on Demand

## Annexos

### ANNEX A. INSTAL·LACIÓ D'ONOS

Abans de començar amb la instal·lació d'ONOS, es requereixen algunes dependències pel seu correcte funcionament. És possible que algunes d'elles ja estiguin instal·lades a la màquina on correrà ONOS, però si no fos el cas, a continuació s'especifiquen les comandes/passos a seguir per instal·lar-les.

- Git:  
sudo apt-get install git
- Zip:  
sudo apt-get install zip
- Unzip:  
sudo apt-get install unzip
- Curl:  
sudo apt-get install curl
- Python (versió 2.7):  
sudo add-apt-repository ppa:deadsnakes/ppa  
sudo apt-get update  
sudo apt-get install python2.7
- Java8:  
sudo apt-get install software-properties-common -y && \  
sudo add-apt-repository ppa:webupd8team/java -y && \  
sudo apt-get update && \  
echo "oracle-java8-installer shared/accepted-oracle-license-v1-1 select true" | sudo debconf-set-selections && \  
sudo apt-get install oracle-java8-installer oracle-java8-set-default -y

Una vegada ja estiguin totes les dependències correctament instal·lades es pot passar a descarregar, instal·lar i configurar ONOS. Per fer-ho s'han de seguir els següents passos:

1. Si es vol utilitzar l'última actualització de la branca master s'ha d'executar la següent comanda: **git clone <https://gerrit.onosproject.org/onos>**. Si pel contrari es vol utilitzar una versió anterior, anar a la pàgina oficial de github d'ONOS (<https://github.com/opennetworkinglab/onos>), seleccionar la branca desitjada i clonar-la o descarregar-la a la màquina.
2. Accedir a la carpeta d'ONOS: **cd onos**.
3. Exportar les variables d'entorn: **export ONOS\_ROOT=\$(pwd)**, on \$(pwd) és el directori on es troba ONOS. (p.e. si onos es troba al directori



/home/ubuntu/onos la comanda a introduir seria: **export ONOS\_ROOT=/home/ubuntu/onos)**

4. Carregar el bash\_profile d'ONOS:  
**source \$ONOS\_ROOT/tools/dev/bash\_profile**
5. Per defecte, ONOS està configurat per córrer a la IP de localhost. Pel nostre escenari s'ha de canviar per la IP privada de la màquina. Per fer-ho s'ha d'obrir el fitxer onos/tools/package/onos-run-karaf amb algun editor de text i a la línia 52 canviar la IP 127.0.0.1 per la IP privada.

```
# Run using the secure SSH client
[ ! -f ~/.ssh/id_rsa.pub ] && ssh-keygen -t rsa -f ~/.ssh/id_rsa -P '' -q
$ONOS_DIR/bin/onos-user-key $(id -un) "$(cut -d\ -f2 ~/.ssh/id_rsa.pub)"
$ONOS_DIR/bin/onos-user-password onos rocks

# Create config/cluster.json (cluster metadata)
IP=${ONOS_IP:-127.0.0.1}
echo "Creating local cluster configs for IP $IP..."
[ -d $ONOS_DIR/config ] || mkdir -p $ONOS_DIR/config
cat > $ONOS_DIR/config/cluster.json <<-EOF
{
  "name": "default",
  "node": {
    "id": "$IP",
    "ip": "$IP",
    "port": 9876
  }
}
EOF
```

**Fig A.1** fitxer de configuració onos-run-karaf

6. Quan ja estiguin aquestes modificacions fetes, a continuació s'ha de compilar tot el codi: **buck build onos**. Aquest procés pot tardar bastant temps en completar-se i es pot donar el cas que la primera i segona vegada que s'executi aparegui algun error. **Si es donés el cas, tornar a executar la mateixa comanda.** Alternativament, per les últimes versions s'ha canviat el sistema de compilació i s'ha passat a utilitzar Bazel. Per tal de compilar el codi amb aquest nou sistema de compilació s'ha d'executar: **bazel build onos**. De la mateixa manera que amb buck, és un procés lent que pot tardar bastant temps i pot donar algun error la primera vegada que s'executi, per tant **si es donés el cas, tornar a executar la mateixa comanda**. Si el problema persisteix, contactar amb el fòrum de discussió [11] o de desenvolupament [12] d'ONOS.

## ANNEX B. INSTAL·LACIÓ D'ATOMIX

Per tal de descarregar, instal·lar i configurar Atomix s'han de seguir els següents passos:

1. Descarregar la versió desitjada (per consultar totes les versions visitar la pàgina <https://atomix.io/downloads/>) executant la següent comanda (exemple amb la versió 3.0.5):

**wget**

**`https://oss.sonatype.org/content/repositories/releases/io/atomix/atomix-dist/3.0.5/atomix-dist-3.0.5.tar.gz`**

2. Extreure els documents del fitxer comprimit que s'ha descarregat al directori desitjat. Per fer-ho s'ha d'executar la següent comanda: **`tar -xzf atomix-dist-3.0.5.tar.gz`**
3. Crear un fitxer amb nom **`pom.xml`** i copiar el següent codi (els camps **version** han de coincidir amb la versió d'Atomix descarregada):

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.upc.atomix</groupId>
  <artifactId>atomix</artifactId>
  <version>1.0-SNAPSHOT</version>

  <dependencies>
    <dependency>
      <groupId>io.atomix</groupId>
      <artifactId>atomix</artifactId>
      <version>3.0.5</version>
    </dependency>
    <dependency>
      <groupId>io.atomix</groupId>
      <artifactId>atomix-raft</artifactId>
      <version>3.0.5</version>
    </dependency>
    <dependency>
      <groupId>io.atomix</groupId>
      <artifactId>atomix-primary-backup</artifactId>
      <version>3.0.5</version>
    </dependency>
    <dependency>
      <groupId>io.atomix</groupId>
      <artifactId>atomix-gossip</artifactId>
      <version>3.0.5</version>
    </dependency>
  </dependencies>

</project>
```

**Fig B.1** fitxer de configuració pom.xml

4. Dins del directori **`atomix/conf/`** modificar el fitxer **`atomix.conf`** amb la configuració dels nodes atomix que hi haurà. A continuació es mostra un exemple de configuració amb 3 nodes Atomix.

```
cluster {
  name: atomix
  discovery {
    type: bootstrap
    nodes.1 {
      id: atomix-1
      address: "10.192.19.111:5679"
    }
    nodes.2 {
      id: atomix-2
      address: "10.192.19.112:5679"
    }
    nodes.3 {
      id: atomix-3
      address: "10.192.19.113:5679"
    }
  }
}

management-group {
  type: raft
  partitions: 1
  storage.level: disk
  members: [atomix-1, atomix-2, atomix-3]
}

partition-groups.raft {
  type: raft
  partitions: 7
  storage.level: disk
  members: [atomix-1, atomix-2, atomix-3]
}
```

**Fig B.2** exemple de fitxer de configuració atomix.conf

5. Fer el build del codi amb la següent comanda (aquesta comanda s'ha d'executar dins del directori que contingui el fitxer **pom.xml**): **mvn clean Package**
6. Fixar les variables d'entorn d'Atomix: **export ATOMIX\_ROOT=./**
7. Arrancar l'agent d'Atomix amb la següent comanda (després de l'opció **-m** s'ha d'indicar l'id que s'hagi posat dins del fitxer atomix.conf pel node actual, i després de l'opció **-a** s'ha d'indicar l'adreça IP del node actual): **bin/atomix-agent -m atomix-1 -a 10.192.19.112:5679**

## ANNEX C. INSTAL·LACIÓ DE BAZEL

Abans de començar amb la instal·lació Bazel, es requereixen algunes dependències pel seu correcte funcionament. És possible que algunes d'elles ja estiguin instal·lades, però si no fos el cas, a continuació s'especifiquen les comandes/passos a seguir per instal·lar-les.

- `sudo apt-get install pkg-config zip g++ zlib1g-dev unzip python`

Una vegada ja estiguin totes les dependències correctament instal·lades es pot passar a descarregar, instal·lar i Bazel. Per fer-ho s'han de seguir els següents passos:

1. Descarregar el fitxer d'instal·lació segons la versió desitjada. (Actualment es recomana utilitzar la versió 0.15.2). Per veure les versions disponibles visitar la pàgina <https://github.com/bazelbuild/bazel/releases>. Per descarregar el fitxer s'ha d'executar la següent comanda (exemple amb la versió 0.15.2) :

```
wget https://github.com/bazelbuild/bazel/releases/download/0.15.2/bazel-0.15.2-installer-linux-x86_64.sh
```

2. Executar el fitxer d'instal·lació de la següent forma:

```
chmod +x bazel-<version>-installer-linux-x86_64.sh
```

```
./bazel-<version>-installer-linux-x86_64.sh --user
```

(L'opció `--user` instal·la Bazel al directori `$HOME/bin` i fixa el path del `.bazelrc` a `$HOME/.bazelrc`. En cas d'algun dubte veure l'ajuda amb l'opció `--help`).

3. Fixar la variable d'entorn: **`export PATH="$PATH:$HOME/bin"`**

## ANNEX D. CONTINGUT DELS FITXERS A MODIFICAR DE L'APLICACIÓ MOVE-DEVICE

Per implementar aquesta nova aplicació s'han de modificar els següents fitxers (l'ordre de modificació no és estricte però es important fer totes les modificacions abans de tornar a compilar el codi, d'altra manera no compilarà correctament):

- **MastershipAdminService.java** (es troba dins del directori `onos/core/api/src/main/java/org/onosproject/mastership/`):

En aquesta interfície s'ha de declarar la nova funció amb els paràmetres `deviceId` de tipus `DeviceId` i `nodeId` de tipus `NodeId` a sota la funció de `balanceRoles()`.

```
59         void balanceRoles();
60
61         void moveDevice(DeviceId deviceId, NodeId nodeId);
```

**Fig D.1** Definició nova funció `moveDevice` dins de `MastershipAdminService.java`.

- **MastershipManager.java** (es troba dins del directori `onos/core/net/src/main/java/org/onosproject/cluster/impl/`):

Aquesta és una de les classes que implementa la interfície anterior, per tant en aquesta classe s'ha d'implementar la funció que s'ha declarat a la interfície.

```
348     @Override
349     public void moveDevice(DeviceId deviceId, NodeId nodeId){
350         ControllerNode controllerNode = clusterService.getNode(nodeId);
351         Map<ControllerNode, Set<DeviceId>> controllerDevices = new HashMap<>();
352
353         Set<DeviceId> devicesOf = new HashSet<>(getDevicesOf(nodeId));
354         controllerDevices.put(controllerNode, devicesOf);
355         Collection<DeviceId> destBucket = controllerDevices.get(controllerNode);
356         Collection<DeviceId> originBucket = new ArrayList();
357
358         originBucket.add(deviceId);
359
360         List<CompletableFuture<Void>> setRoleFutures = Lists.newLinkedList();
361         changeMastership(controllerNode, destBucket, originBucket, count: 1, setRoleFutures);
362     }
363 }
```

**Fig D.2** Implementació nova funció `moveDevice` dins de `MastershipManager.java`.

- **VirtualNetworkMastershipManager.java** (es troba dins del directori `onos/incubator/net/src/main/java/org/onosproject/incubator/net/virtual/impl/`).

Aquesta classe també implementa la interfície anterior, per tant també s'ha d'implementar la funció en aquesta classe encara que aquest projecte no l'utilitza. D'altra manera el codi no compilaria.

```

200      @Override
201      public void moveDevice(DeviceId deviceId, NodeId nodeId){
202          log.info("moveDevice function inside VirtualNetworkMastershipManager.");
203      }

```

**Fig D.3** Implementació nova funció `moveDevice` dins de `VirtualNetworkMastershipManager.java`.

- Afegir una nova classe amb nom **MoveDeviceToNodeCommand** dins de `onos/cli/src/main/java/org/onosproject/cli/`:

Aquesta classe és on s'hi defineix l'aplicació `move-device` que crida la funció declarada a la interfície **MastershipAdminService.java** i implementada a la classe **MastershipManager.java**

```

1  // ...
16
17  package org.onosproject.cli;
18
19  import org.apache.karaf.shell.commands.Argument;
20  import org.apache.karaf.shell.commands.Command;
21  import org.onosproject.cluster.NodeId;
22  import org.onosproject.mastership.MastershipAdminService;
23  import org.onosproject.net.DeviceId;
24
25  // ...
26  // Forces device mastership rebalancing.
27  // ...
28  @Command(scope = "onos", name = "move-device",
29           description = "Forces device move to node" +
30           "")
31  public class MoveDeviceToNodeCommand extends AbstractShellCommand {
32
33      @Argument(index = 0, name = "device", description = "Device pretending to move",
34               required = true, multiValued = false)
35      String device = null;
36
37      @Argument(index = 1, name = "controller", description = "destination controller",
38               required = true, multiValued = false)
39      String controller = null;
40
41      @Override
42      protected void execute()
43      {
44          DeviceId deviceId = DeviceId.deviceId(device);
45          NodeId controllerId = NodeId.nodeId(controller);
46          get(MastershipAdminService.class).moveDevice(deviceId, controllerId);
47      }
48
49  }

```

**Fig D.4** Nou fitxer `MoveDeviceToNodeCommand.java`

- **shell-config.xml** (es troba dins del directori **onos/cli/src/main/resources/OSGI-INF/blueprint/**)

Dins d'aquest fitxer es defineixen totes les comandes que es poden executar des del CLI, per tant s'ha d'introduir una nova comanda per utilitzar la nova aplicació.

```
114         <action class="org.onosproject.cli.MoveDeviceToNodeCommand"/>
115         <completers>
116             <ref component-id="deviceIdCompleter"/>
117             <ref component-id="nodeIdCompleter"/>
118         </completers>
119     </command>
```

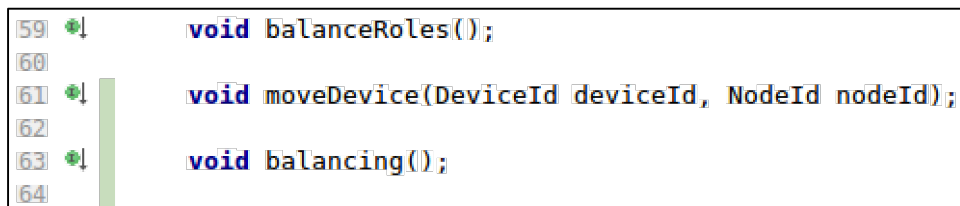
**Fig D.5** Definició nova aplicació per utilitzar-la al CLI.

## ANNEX E. CONTINGUT DELS FITXERS A MODIFICAR DE L'APLICACIÓ BALANCING

Per implementar aquesta nova funció, per la part del CLI, s'han de modificar els mateixos fitxers que per l'aplicació de move-device. A continuació s'expliquen les modificacions al codi per aquesta nova funció (igual que abans, l'ordre de modificació no és estricte però es important fer totes les modificacions abans de tornar a compilar el codi, d'altra manera no compilarà correctament):

- **MastershipAdminService.java** (es troba dins del directori `onos/core/api/src/main/java/org/onosproject/mastership/`):

En aquesta interfície s'ha de declarar la nova funció a sota la funció de `moveDevice(DeviceId deviceId, NodeId nodeId)`.



```
59 void balanceRoles();
60
61 void moveDevice(DeviceId deviceId, NodeId nodeId);
62
63 void balancing();
64
```

**Fig E.1** Definició nova funció balancing dins de MastershipAdminService.java.

- **MastershipManager.java** (es troba dins del directori `onos/core/net/src/main/java/org/onosproject/cluster/impl/`):

Aquesta és una de les classes que implementa la interfície anterior, per tant en aquesta classe s'ha d'implementar la funció que s'ha declarat a la interfície.



```

291     int count = 1;
292
293     @Override
294     public void balancing(){
295
296         List<ControllerNode> nodes = new ArrayList(clusterService.getNodes());
297
298         Map<ControllerNode, Set<DeviceId>> controllerDevices = new HashMap<>();
299
300         for (ControllerNode node : nodes) {
301             Set<DeviceId> devicesOf = new HashSet<>(getDevicesOf(node.id()));
302             if (clusterService.getState(node.id()).isActive()) {
303                 controllerDevices.put(node, devicesOf);
304             }
305         }
306
307         ControllerNode controllerNode = null;
308         Collection<DeviceId> destBucket = new ArrayList<>();
309         Collection<DeviceId> originBucket = new ArrayList<>();
310
311         if (count == 1){
312             controllerNode = nodes.get(1);
313             destBucket.addAll(controllerDevices.get(nodes.get(1)));
314             originBucket.addAll(controllerDevices.get(nodes.get(0)));
315
316             count++;
317         } else if (count == 2){
318             controllerNode = nodes.get(2);
319             destBucket.addAll(controllerDevices.get(nodes.get(2)));
320             originBucket.addAll(controllerDevices.get(nodes.get(1)));
321             count++;
322         } else if (count == 3){
323             controllerNode = nodes.get(0);
324             destBucket.addAll(controllerDevices.get(nodes.get(0)));
325             originBucket.addAll(controllerDevices.get(nodes.get(2)));
326             count = 1;
327         }
328
329         List<CompletableFuture<Void>> setRoleFutures = Lists.newLinkedList();
330
331         changeMastership(controllerNode, destBucket, originBucket, count, setRoleFutures);
332     }

```

Fig E.2 Implementació nova funció balancing dins de MastershipManager.java.

- **VirtualNetworkMastershipManager.java** (es troba dins del directori `onos/incubator/net/src/main/java/org/onosproject/incubator/net/virtual/impl/`).

Aquesta classe també implementa la interfície anterior, per tant també s'ha d'implementar la funció en aquesta classe encara que aquest projecte no l'utilitza. D'altra manera el codi no compilaria.

```

205     @Override
206     public void balancing(){
207         log.info("balancing function inside VirtualNetworkMastershipManager.");
208     }

```

Fig E.3 Implementació nova funció balancing dins de VirtualNetworkMastershipManager.java.

- Afegir una nova classe amb nom **BalancingCommand** dins de **onos/cli/src/main/java/org/onosproject/cli**:

Aquesta classe és on s'hi defineix l'aplicació balancing que crida la funció declarada a la interfície **MastershipAdminService.java** i implementada a la classe **MastershipManager.java**

```

1  /**
16 package org.onosproject.cli;
17
18 import org.apache.karaf.shell.commands.Command;
19 import org.onosproject.mastership.MastershipAdminService;
20
21 /**
22  * Forces device mastership rebalancing.
23  */
24 @Command(scope = "onos", name = "balancing",
25           description = "Forces rebalancing")
26 public class BalancingCommand extends AbstractShellCommand {
27
28     @Override
29     protected void execute() { get(MastershipAdminService.class).balancing(); }
30
31 }

```

**Fig E.4** Nou fitxer BalancingCommand.java

- **shell-config.xml** (es troba dins del directori **onos/cli/src/main/resources/OSGI-INF/blueprint/**)

Dins d'aquest fitxer es defineixen totes les comandes que es poden executar des del CLI, per tant s'ha d'introduir una nova comanda per utilitzar la nova aplicació.

```

120 <command>
121     <action class="org.onosproject.cli.BalancingCommand"/>
122 </command>

```

**Fig E.5** Definició nova aplicació per utilitzar-la al CLI.

Per la part de l'execució periòdica les modificacions per les versions 1.14 i 1.15 són diferents. Per la versió 1.14 s'han realitzar les següents modificacions:

Dins del directori **onos/apps/** s'ha d'afegir un nou arbre de directoris: **/balancing/src/main/java/org/onosproject/balancing/**, i aquí dins s'han de crear dos fitxers nous, **Balancing.java** i **package-info.java**



**Fig E.6** Creació del nou arbre de directoris i fitxers dins de onos/apps/.

Quan aquests fitxers ja estiguin creats s'hi ha de posar el següent codi dins:

```

1  /*
2   * Copyright 2015-present Open Networking Foundation
3   *
4   * Licensed under the Apache License, Version 2.0 (the "License");
5   * you may not use this file except in compliance with the License.
6   * You may obtain a copy of the License at
7   *
8   * http://www.apache.org/licenses/LICENSE-2.0
9   *
10  * Unless required by applicable law or agreed to in writing, software
11  * distributed under the License is distributed on an "AS IS" BASIS,
12  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13  * See the License for the specific language governing permissions and
14  * limitations under the License.
15  */
16
17  /**
18   * Balancing.
19   */
20  package org.onosproject.balancing;

```

**Fig E.7** Fitxer package-ingo.java.

Pel fitxer Balancing.java són necessaris els següents imports:

```

1  //.../
16
17  package org.onosproject.balancing;
18
19  import org.apache.felix.scr.annotations.Activate;
20  import org.apache.felix.scr.annotations.Component;
21  import org.apache.felix.scr.annotations.Deactivate;
22  import org.apache.felix.scr.annotations.Modified;
23  import org.apache.felix.scr.annotations.Property;
24  import org.apache.felix.scr.annotations.Reference;
25  import org.apache.felix.scr.annotations.ReferenceCardinality;
26  import org.onlab.util.Tools;
27  import org.onosproject.cfg.ComponentConfigService;
28  import org.onosproject.cluster.ClusterService;
29  import org.onosproject.cluster.LeadershipEvent;
30  import org.onosproject.cluster.LeadershipEventListener;
31  import org.onosproject.cluster.LeadershipService;
32  import org.onosproject.cluster.NodeId;
33  import org.onosproject.mastership.MastershipAdminService;
34  import org.onosproject.mastership.MastershipEvent;
35  import org.onosproject.mastership.MastershipListener;
36  import org.onosproject.mastership.MastershipService;
37  import org.osgi.service.component.ComponentContext;
38  import org.onosproject.net.region.RegionEvent;
39  import org.onosproject.net.region.RegionListener;
40  import org.onosproject.net.region.RegionService;
41  import org.slf4j.Logger;
42
43  import java.util.Dictionary;
44  import java.util.concurrent.Executors;
45  import java.util.concurrent.Future;
46  import java.util.concurrent.ScheduledExecutorService;
47  import java.util.concurrent.TimeUnit;
48  import java.util.concurrent.atomic.AtomicBoolean;
49  import java.util.concurrent.atomic.AtomicReference;
50
51  import static org.onlab.util.Tools.groupedThreads;
52  import static org.slf4j.LoggerFactory.getLogger;
53
54  //...//
55
56  @Component(immediate = true)
57  public class Balancing {
58
59
60
61
62

```

Fig E.8 Imports fitxer Balancing.java.

Quan ja es tinguin els imports i la definició de la classe, s'han de copiar totes les funcions de la classe **MastershipLoadBalancer.java** que es troba dins la aplicació **mlb** (directori **onos/apps/mlb/src/main/java/org/onosproject/mlb**). A continuació, dins de la funció **public void run()**, tal i com es mostra a la següent imatge, s'ha de canviar **mastershipAdminService.balanceRoles()** per **mastershipAdminService.balancing()**:

```

175     private class BalanceTask implements Runnable {
176
177         @Override
178         public void run() {
179             // nextTask is now running, free the spot so that it is possible
180             // to queue up another upcoming task.
181             nextTask.set(null);
182
183             log.info("\nbalancing app\n");
184             mastershipAdminService.balancing();
185             log.info("Completed balance roles");
186         }
187     }

```

**Fig E.9** Execució funció balancing() dins del fitxer Balancing.java.

- Dins del directori **onos/apps/balancing/** crear un fitxer amb nom **BUCK** (serveix per poder compilar correctament l'aplicació) i posar-hi el següent codi:

```

1  COMPILE_DEPS = [
2      '//lib:core_deps',
3  ]
4
5  osgi_jar_with_tests (
6      deps = COMPILE_DEPS,
7  )
8
9  onos_app (
10     title = 'Balancing',
11     category = 'Utility',
12     url = 'http://onosproject.org',
13     description = 'Balance devices.',
14 )

```

**Fig E.10** Fitxer BUCK de la aplicació balancing.

- Dins del fitxer **modules.defs** (es troba dins del directori **onos/**), a la part on es defineixen els **ONOS\_APPS** afegir una nova línia amb la informació de l'aplicació balancing seguint el mateix format de la resta d'aplicacions.

```

149 ONOS_APPS = [
150     # Apps
151     '//apps/dhcp:onos-apps-dhcp-oar',
152     '//apps/dhcprelay:onos-apps-dhcprelay-oar',
153     '//apps/fwd:onos-apps-fwd-oar',
154     '//apps/packet-stats:onos-apps-packet-stats-oar',
155     '//apps/acl:onos-apps-acl-oar',
156     '//apps/bgprouter:onos-apps-bgprouter-oar',
157     '//apps/cip:onos-apps-cip-oar',
158     '//apps/drivermatrix:onos-apps-drivermatrix-oar',
159     '//apps/events:onos-apps-events-oar',
160     '//apps/proxyarp:onos-apps-proxyarp-oar',
161     '//apps/segmentrouting:onos-apps-segmentrouting-oar',
162     '//apps/gangliametrics:onos-apps-gangliametrics-oar',
163     '//apps/graphitemetrics:onos-apps-graphitemetrics-oar',
164     '//apps/flowanalyzer:onos-apps-flowanalyzer-oar',
165     '//apps/intentsync:onos-apps-intentsync-oar',
166     '//apps/influxdbmetrics:onos-apps-influxdbmetrics-oar',
167     '//apps/metrics:onos-apps-metrics-oar',
168     '//apps/mfwd:onos-apps-mfwd-oar',
169     '//apps/mlb:onos-apps-mlb-oar',
170     '//apps/balancing:onos-apps-balancing-oar',

```

**Fig E.11** Definició de la aplicació balancing dins del fitxer modules.defs.

Per la versió 1.15 s'han realitzar les següents modificacions:

La creació de l'arbre de directoris **/balancing/src/main/java/org/onosproject/balancing/** dins de **onos/apps/**, i la creació dels fitxers **Balancing.java** i **package-info.java** i el seu contingut segueix sent idèntic que per la versió 1.14. La diferència és que aquest cop enlloc de crear dins del directori **onos/apps/balancing/** un fitxer de nom **BUCK**, se n'ha de crear un amb nom **BUILD** amb el següent contingut:

```

1  osgi_jar_with_tests(
2      deps = CORE_DEPS,
3  )
4
5  onos_app(
6      category = "Utility",
7      description = "Balance devices.",
8      title = "Balancing",
9      url = "http://onosproject.org",
10 )

```

**Fig E.12** Fitxer BUILD de l'aplicació balancing.

L'altra diferència és que dins del fitxer **modules.bzl** (enlloc del fitxer **modules.defs**, també situat dins del directori **onos/**), on es defineixen els **ONOS\_APPS** s'ha d'afegir una nova línia amb la informació de l'aplicació balancing.

```
148     ONOS_APPS = [  
149         # Apps  
150         "//apps/dhcp:onos-apps-dhcp-oar",  
151         "//apps/dhcprelay:onos-apps-dhcprelay-oar",  
152         "//apps/fwd:onos-apps-fwd-oar",  
153         "//apps/packet-stats:onos-apps-packet-stats-oar",  
154         "//apps/acl:onos-apps-acl-oar",  
155         "//apps/bgprouter:onos-apps-bgprouter-oar",  
156         "//apps/cip:onos-apps-cip-oar",  
157         "//apps/drivermatrix:onos-apps-drivermatrix-oar",  
158         "//apps/events:onos-apps-events-oar",  
159         "//apps/proxyarp:onos-apps-proxyarp-oar",  
160         "//apps/segmentrouting:onos-apps-segmentrouting-oar",  
161         "//apps/gangliametrics:onos-apps-gangliametrics-oar",  
162         "//apps/graphitemetrics:onos-apps-graphitemetrics-oar",  
163         "//apps/flowanalyzer:onos-apps-flowanalyzer-oar",  
164         "//apps/intentsync:onos-apps-intentsync-oar",  
165         "//apps/influxdbmetrics:onos-apps-influxdbmetrics-oar",  
166         "//apps/metrics:onos-apps-metrics-oar",  
167         "//apps/mfwd:onos-apps-mfwd-oar",  
168         "//apps/mlb:onos-apps-mlb-oar",  
169         "//apps/balancing:onos-apps-balancing-oar",
```

**Fig E.13** Definició de la aplicació balancing dins del fitxer modules.bzl.